HUNTON & WILLIAMS

1900 K STREET, N.W.
WASHINGTON, D.C. 20006-1109

TEL 202 • 955 • 1500
FAX 202 • 778 • 2201

SCOTT D. BALDERSTON
DIRECT DIAL: 202 • 955 • 1935
EMAIL: sbalderston@hunton.com

FILE NO: 53470.000038

September 13, 2000

21967
PATENT TRADEMARK OFFICE

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

Re: Filing of New U.S. Utility Patent Application
Title: **System And Method For Voice-Enabled Input For Use In The Creation And Automatic Deployment Of Personalized, Dynamic And Interactive Voice Services**
Inventors: **Hannes EBERLE, Christopher S. LEON, Bodo MAASS, Anurag PATNAIK, Alberto SANTA ANA and Michael ZIRNGIBL**

Dear Sir:

Attached is a new patent application for filing in the United States Patent and Trademark Office including sixty (60) pages of Specification, three (3) pages of Claims (numbered 1—20), one (1) page Abstract, eleven (11) sheets of Drawings (labeled Figs. 1—10), an unexecuted Joint Declaration and an Information Disclosure Statement.

This application claims priority of U.S. Provisional Application Serial No. 60/153,222, filed September 13, 1999.

The filing fee is calculated as follows:

| | | | | |
|---|---|---|---|---|
| | | | | *AMOUNT* |
| | | *BASIC FILING FEE* | | **$690.00** |
| | *No. of Claims* | *No. in Excess* | *Rate* | |
| *Number of Claims in Excess of:* **20** | 20 | 0 | $18.00 | **.00** |
| *Independent Claims in Excess of:* **3** | 2 | 0 | $78.00 | **.00** |
| *First Presentation of Multiple Dependent Claims* | | | $ 130.00 | |
| *Reduce by ½ for Small Entity* | | | | |
| *Recordation of Assignment* | | | | |
| | | **TOTAL FEE DUE** | | **$ 690.00** |

HUNTON & WILLIAMS

September 13, 2000
Page 2

A check in the amount of ***$690.00*** is attached to cover the basic application filing fee. In the event of any variance between the amount enclosed and the Patent and Trademark Office charges, please charge or credit any difference to the undersigned's Deposit Account No. 50-0206.

Please direct all communication concerning this application to:

James G. Gatto, Esq.
Hunton & Williams
1900 K Street, N.W.
Suite 1200
Washington, DC 20006

Respectfully submitted,

HUNTON & WILLIAMS

By: Scott D Balderston

Scott D. Balderston
Registration No. 35,436

1900 K Street, NW, Suite 1200
Washington, D.C. 20006-1109
Telephone: (202) 955-1500
Facsimile: (202) 778-2201

Dated: September 13, 2000

# System and Method for Voice-Enabled Input For Use in the Creation and Automatic Deployment of Personalized, Dynamic and Interactive Voice Services

## Field of the Invention

This invention relates to a system and method for creation and automatic deployment of personalized, dynamic and interactive voice services accepting voice input, including information derived from on-line analytical processing (OLAP) systems. More specifically, the invention relates to a system and method that enable personalized delivery of information in real-time, via two-way natural language voice communication with a voice-enabled terminal device. The system and method combines personalized information broadcast technology with a calling platform, a text-to-speech (TTS) engine and the structure that creates telecasts—two-way communication between information consumers and database (or other) applications, relevant to a subscriber and delivered by a call server. A unique telecast is generated for each subscriber scheduled to receive voice service content. A voice server contains the call structure and data, voice style parameters for the subscriber and personal identification information designated for the subscriber. The invention accepts spoken commands and requests, and may authenticate the recipient on the basis of a voice profile.

## Background of the Invention

The ability to act quickly and decisively in today's increasingly competitive marketplace is critical to the success of any organization. The volume of data that is available to organizations is rapidly increasing and frequently overwhelming. The availability of large volumes of data presents various challenges. One challenge is to avoid inundating an individual

with unnecessary information. Another challenge is to ensure all relevant information is available in a timely manner.

One known approach to addressing these and other challenges is known as data warehousing. Data warehouses, relational databases, and data marts are becoming important elements of many information delivery systems because they provide a central location where a reconciled version of data extracted from a wide variety of operational systems may be stored. As used herein, a data warehouse should be understood to be an informational database that stores shareable data from one or more operational databases of record, such as one or more transaction-based database systems. A data warehouse typically allows users to tap into a business's vast store of operational data to track and respond to business trends that facilitate forecasting and planning efforts. A data mart may be considered to be a type of data warehouse that focuses on a particular business segment.

Decision support systems have been developed to efficiently retrieve selected information from data warehouses. One type of decision support system is known as an on-line analytical processing system. In general, OLAP systems analyze the data from a number of different perspectives and support complex analyses against large input data sets.

There are at least three different types of OLAP architectures - ROLAP, MOLAP, and HOLAP. ROLAP ("Relational On-Line Analytical Processing") systems are systems that use a dynamic server connected to a relational database system. Multidimensional OLAP ("MOLAP") utilizes a proprietary multidimensional database ("MDDB") to provide OLAP analyses. The main premise of this architecture is that data must be stored multi-dimensionally to be viewed

multi-dimensionally. A HOLAP ("Hybrid On-Line Analytical Processing") system is a hybrid of these two.

ROLAP is a three-tier, client/server architecture comprising a presentation tier, an application logic tier and a relational database tier. The relational database tier stores data and connects to the application logic tier. The application logic tier comprises a ROLAP engine that executes multidimensional reports from multiple end users. The ROLAP engine integrates with a variety of presentation layers, through which users perform OLAP analyses. The presentation layers enable users to provide requests to the ROLAP engine. The premise of ROLAP is that OLAP capabilities are best provided directly against a relational database, *e.g.*, the data warehouse.

In a ROLAP system, data from transaction-processing systems is loaded into a defined data model in the data warehouse. Database routines are run to aggregate the data, if required by the data model. Indices are then created to optimize query access times. End users submit multidimensional analyses to the ROLAP engine, which then dynamically transforms the requests into SQL execution plans. The SQL is submitted to the relational database for processing, the relational query results are cross-tabulated, and a multidimensional result set is returned to the end user. ROLAP is a fully dynamic architecture capable of utilizing pre-calculated results when they are available, or dynamically generating results from atomic information when necessary.

The ROLAP architecture directly accesses data from data warehouses, and therefore supports optimization techniques to meet batch window requirements and to provide fast

response times. These optimization techniques typically include application-level table partitioning, aggregate inferencing, denormalization support, and multiple fact table joins.

MOLAP is a two-tier, client/server architecture. In this architecture, the MDDB serves as both the database layer and the application logic layer. In the database layer, the MDDB system is responsible for all data storage, access, and retrieval processes. In the application logic layer, the MDDB is responsible for the execution of all OLAP requests. The presentation layer integrates with the application logic layer and provides an interface through which the end users view and request OLAP analyses. The client/server architecture allows multiple users to access the multidimensional database.

Information from a variety of transaction-processing systems is loaded into the MDDB System through a series of batch routines. Once this atomic data has been loaded into the MDDB, the general approach is to perform a series of batch calculations to aggregate along the orthogonal dimensions and fill the MDDB array structures. For example, revenue figures for all of the stores in a state would be added together to fill the state level cells in the database. After the array structure in the database has been filled, indices are created and hashing algorithms are used to improve query access times.

Once this compilation process has been completed, the MDDB is ready for use. Users request OLAP reports through the presentation layer, and the application logic layer of the MDDB retrieves the stored data.

The MOLAP architecture is a compilation-intensive architecture. It principally reads the pre-compiled data, and has limited capabilities to dynamically create aggregations or to calculate business metrics that have not been pre-calculated and stored.

The hybrid OLAP ("HOLAP") solution is a mix of MOLAP and relational architectures that support inquiries against summary and transaction data in an integrated fashion. The HOLAP approach enables a user to perform multidimensional analysis on data in the MDDB. However, if the user reaches the bottom of the multidimensional hierarchy and requires more detailed data, the HOLAP engine generates an SQL to retrieve the detailed data from the source relational database management system ("RDBMS") and returns it to the end user. HOLAP implementations rely on simple SQL statements to pull large quantities of data into the mid-tier, multidimensional engine for processing. This constrains the range of inquiry and returns large, unrefined result sets that can overwhelm networks with limited bandwidth.

As described above, each of these types of OLAP systems are typically client-server systems. The OLAP engine resides on the server side and a module is typically provided at a client-side to enable users to input queries and report requests to the OLAP engine. Many current client-side modules are typically stand alone software modules that are loaded on client-side computer systems. These systems require that a user must learn how to operate the client-side software module in order to initiate queries and generate reports.

Additionally, the product DSS Broadcaster was introduced by Microstategy to output OLAP reports to users via various output delivery devises. The current version of DSS Broadcaster is a one-way delivery of information from the DSS Broadcaster system to the user.

Another system in use today is a interactive telephone system that enables user to interactively request information through a computerized interface. These systems require that the user call in to a central number to access the system and request information by stepping

through various options in predefined menu choices. Such information may include accessing account information, movie times, service requests, etc.

Another problem with these systems is that the menu structure is typically set and not customized to a particular's users preferences or customized to the information available to that user. Therefore, a user may have to wade through a host of inapplicable options to get to the one or two options applicable to that user. Further, a user may be interested in a particular report. With existing telephone call-in systems, that user has to input the same series of options each time they want to hear the results of that report. If the user desires to run that report frequently, the telephone input system described is a very time consuming and wasteful method of accessing that information. Also, if a particular user may only be interested in knowing if a particular value or set of values in the report has changed over a predetermined period of time, in such a system, the user would be required to initiate the report frequently and then scan through the new report to determine if the information has changed over the time period specified.

Further, reports may be extensive and may contain a large amount of information for a user to sort through each time a report is run. Therefore, the user may have to wait a long time for the report to be generated once they input the appropriate parameters for the report.

Moreover, the delivery of voice messaging services having a large number of prompted menus can be cumbersome to respond to using keypad input, which is also susceptible to mistaken key strikes, broken keys and other inconveniences.

These and other drawbacks exist with current OLAP interface systems.

**Summary of Invention**

One aspect of the invention relates to the delivery of voice message service, with closed loop feedback from the user including voice input commands. The subscriber may, for instance, receive a voice service message concerning a financial account, and be queried whether they wish to buy, sell or perform another transaction on that account. The subscriber's responses can be spoken commands, which the call server receives and interprets to execute a service or present a further voice menu. Because the interactive input may be completely in the form of audible instructions, errors such as missed keypad entries are avoided.

The subscriber can hear and respond to voice service alerts even when keypad input is difficult or impractical, such as when the user is working in poor lighting. The naturalness and practicality of the interface is therefore enhanced. Likewise, the range of input choices can be increased without necessarily increasing the depth of a voice menu tree, because possible subscriber responses are not limited to the 12 keys of a telephone keypad.

The invention in another regard may be adapted for use in more than one language, by substituting voice recognition modules tailored to other languages or dialects.

The invention in another regard may use the characteristics of a recipient's voice input, such as cadence, gender, spectral signature and others to authenticate the recipient as the intended subscriber or otherwise.

Another aspect of the invention relates to the creation and deployment of personalized, dynamic and interactive voice services including delivery of information from OLAP systems or other data repositories. A voice service comprises at least six characteristics: service type, content type, schedule, content definition, personalization settings and error handling settings.

Once a voice service is set-up, a user can subscribe to the voice service. Based upon the occurrence of a predetermined event (for example, a schedule and/or a trigger event) a voice service may be executed. Execution of a voice service preferably comprises the steps of generating voice service content, creating an active voice page (AVP) (call structure), sending the AVP to a service queue, conducting an interactive voice broadcast (IVB) and writing user responses. Preferably during each IVB, a synthesized, natural-sounding voice greets the recipient by name, identifies itself and provides information relevant to the subscriber. The system then either prompts the person for a response or presents a choice of menu options that lead to more dialog.

One aspect of the invention combines elements of data analysis, web and telephony technologies into one solution. The result is a truly personalized voice broadcasting server. Unlike other telephony applications, the system of the present invention is pro-active, dynamic, personalized and interactive.

Unlike traditional call centers and Interactive Voice Response (IVR) systems which require a user to call a specific number and then spend a significant amount of time searching through the information presented, the present invention can automatically generate a fully personalized message and automatically make an outbound call. Calls maybe based on individually defined schedules or exception criteria, which are usually submitted by the user during a subscription process (*e.g.*, web- or phone-based).

The system creates fully personalized calls by generating an XML-based document (the AVP) for every message that goes out. This document then determines in real-time the individual call flow. Call content varies depending on the user and the responses that are entered. By

simply using the telephone keypad (or other input mechanism) users can control the entire call flow, select given options, enter information at user prompts and conduct transactions. Additionally, the system can collect user inputs and provide them to external applications, databases and transaction systems.

For increased flexibility in application development and for easy integration with external applications, one embodiment of the invention uses an XML/XSL based approach for its execution of an IVB. Extensible Markup Language (XML) is a data tagging language that allows the system to turn data from multiple sources into a system independent format that can easily be processed and formatted. By applying Extensible Stylesheet Language (XSL) documents to this XML-based information, the system defines in real-time how information will be transformed to generate personal reports in natural language format, as well as to determine data driven call-flows. Telecaster Markup Language (TML) is a derivative of XML. TML is designed specifically for text-to-speech conversion and interactive voice broadcasting.

Active Voice Pages represent a novel concept for personalizing phone-based user dialogs. Active Voice Pages are XML/TML based documents that are dynamically generated by the system and determine every aspect of the individual user interaction.

Active Voice Pages may include one or more of the following features: personalized reports including personal greetings with the recipient's name; personalized and fully data-driven menu options; fully encrypted transmission of personal PIN code information; extensive error handling capabilities such as user time-out; comprehensive interaction-driven call logic, *e.g.*, number of retries for entering PIN code information; XML-based scripting language for storing

user inputs in pre-defined variables, conducting real-time calculations, etc.; and flags to initiate communication and data transfer to external applications

The invention further enables and takes advantage of the closed-loop capability of standard telephones (and other voice enabled terminal devices). By simply pressing the keys of their phone users can instantly initiate feedback and conduct transactions. As soon as a call is initiated, the system's embedded telephony modules monitor the call line and transform user inputs (coming in as touchtone signals) into textual and numerical information that can be processed by the current Active Voice Page. When a user is prompted to enter information, the response updates a pre-defined variable stored in the Active Voice Page. By combining this information with data about the user and the current call session, a transaction can be uniquely identified and securely transmitted to external applications, databases or transaction servers.

As described above, the system stores user inputs into the current Active Voice Page. In addition to commands used to determine call content, structure and user prompts, active voice pages can also contain flags to initiate external programs. As soon as the active voice page reaches this flag during the call flow, the program is initiated and pre-determined variables are transferred to external applications. Advantageously, the information already exists in a standardized format (XML) to facilitate integration. For example, the system can execute SQL statements, make direct API calls or transfer information via XML-based data and application gateways.

The voice services are based on real-time, text-to-speech conversion, to allow for truly personalized messaging. Unlike other telephony applications such as phone-banking or phone-based trading services that traditionally use static pre-recorded files, text-to-speech conversion

does not impose any limitations on the content of the call. The system can speak customer names, product brands and other terms. Further, it leverages specific algorithms to resolve special content such as numbers, phone numbers or dates etc. In certain cases, it may be beneficial to include pre-recorded dialog components and other audio files (sound effects, music, testimonials, etc.). Thus, the system enables blending of static and dynamic content in various parts of the message.

According to one embodiment, the call server of the present invention comprises computer telephony software that processes each call, TML/XML parser for resolving the Active Voice Pages, a text-to-speech engine, and a call table where statistics and responses are stored.

An enterprise may deploy thousands of data-driven telephone calls designed to enhance business processes, sell information, or market products and services. Each telephone call can be customized to target the specific needs and interests of the user (subscriber). The structure of each call is entirely dynamic, driven by current data values and presenting a different experience to each user.

Voice service content is comprised of messages to be played to the recipient and prompts designed to retrieve information from the recipient. The voice service interfaces allow the administrator to define a series of dialogs that contain messages and requests for user inputs and determine the call flow between these dialogs based on selections made by the user.

A Voice Service can be defined using a Voice Service wizard. There are at least two ways to create voice service content within the Voice Service Wizard: By using the Dialog Wizard, which steps the administrator through the various components of a dialog; and by using

the Voice Content Editor, which allows the administrator to simulate the execution of a call and to view all dialogs while inserting content directly into the dialog tree structure.

The administrator builds a voice service by defining the structure of an IVB and adding content. The IVB structure (or call structure) determines how the Call System and the user will interact during a call. During a typical call, information is exchanged between the Call System and the user. The Call System reads a message to the user and, in response, the user may press buttons on a telephone touch pad dial to hear more information. This shapes the structure of a basic exchange between the call system and the user. The user's responses may be stored during the call and processed during the call or after by the system or other applications.

The Voice Service Wizard provides an easy-to-use interface for the creation of a voice service. It can also be used to update and edit existing voice services. The wizard starts with an introduction screen that describes the steps to be followed.

The Voice Service Wizard allows the administrator to give the service a name and description; set the service's schedule or trigger event(s); define the service's content; select personalization modes for each project; and specify how to handle error cases. The structure of a service as defined in the Voice Service Wizard is translated into TML before being sent to a call server. The Voice Service Wizard uses TML tags that define the elements of an IVB.

The DIALOG element defines a unit of interaction between the user and the system. It can contain all of the other elements. A Dialog element preferably cannot be contained in another element, including another Dialog. The SPEECH element defines the text to be read to the user. The PROMPT element is used to define that a sequence of keypresses is expected as response. The INPUT element defines a section of a Dialog that contains interactive elements

*i.e.*, those elements that pertain to response expected from a user and its validation. The OPTION element defines a menu choice that is associated with a particular phone key. The FOR-EACH element loops through a list of variables, *e.g.*, contained in a database report, or from user input to dynamically generate speech from data. The ERROR element is a child of an INPUT element. It defines the behavior of the Call System, if a subscriber makes an invalid response, such as pressing a key that has not been associated with any choice by an option, or entering input that does not meet the filter criteria of a PROMPT statement. SYS-ERROR system defines a handler for certain call system events such as expiration of the waiting time for user input.

In addition to the elements described above, voice services may also use at least two other features to enhance the administrator's ability to design powerful voice services.

Call Flow Reports allow the administrator to generate the structure of a call based on data returned by a report or query. For instance, the options of a dialog may correspond to the rows returned by a personalized report. The report data is converted to options by application of an XSL, and inserted into the static structure of the call when the service is executed. XSL Style sheets can be associated with Call Flow Reports or with Content reports. With Call Flow reports, they are used to generate the call structure by creating dialogs, options and prompts based on report data. With content reports, they are used to generate a plain text string that may comprise the message at any node in the call structure.

In creating a voice service the administrator structures the flow of an IVB by defining a number of dialogs and the conditions that will lead the recipient from one dialog to another. A dialog is created by adding elements to a telecast tree, with the highest level element being the

dialog element. The dialog can consist of multiple speech sections and an input section that contains option or prompt or error elements. A dialog also contains an error node that defines how errors made by recipients should be handled.

According to another embodiment, the system and method of the present invention may comprise a voice service bureau. According to one embodiment, the voice service bureau (VSB) accepts call requests from remotely located client servers via secured HTTPS protocol, authenticates the requests and then makes the calls to the subscribers. If the invention is deployed in a VSB mode, the call server may reside at a VSB location (*e.g.*, remote from the client server). The VSB is a network operations center constructed and maintained for the purpose of processing telephone calls requested by a remote broadcast server, *e.g.*, a MicroStrategy Broadcast Server. The VSB may receive call requests via the Internet.

The VSB enables a customer to access complete voice and audio broadcasting functionality without being required to purchase or maintain any additional telephone lines, telephony hardware or calling software. To use the VSB, a customer may be required to establish a VSB account and create voice services. No further administrative duties are required by the customer to use the VSB. The VSB provides complete system and network administration services for its proper operation and maintenance.

According to another embodiment, a VSB may include the functionality necessary to create voice services as well. In this embodiment, a user could subscribe to voice services *e.g.*, via phone or web, and receive IVBs without maintaining any infrastructure.

According to another aspect of the invention, the content and structure of a voice service is defined in TML format and sent to the call server. TML follows the Extended Markup

Language (XML) syntax and is used to tag distinct parts of a telephone interaction that are required in order to deliver and/or prompt users with information.

Inbound calling can be handled in a variety of ways. According to one embodiment the system could create personalized Active Voice Pages in batch load, create a mechanism that identifies an inbound caller and redirects him to his personal AVP. AVPs could be stored and managed using a RDBMS system using text fields or a "BLOB" approach. Even without personalization, TML is flexible enough to support inbound calling and IVR systems. According to another embodiment, the system enables an inbound caller to search for a particular AVP, *e.g.*, be entering alpha-numeric characters using the telephone keypad.

Other features and advantages of the present invention will be apparent to one of ordinary skill in the art upon reviewing the detailed description of the present invention.

**Brief Description of the Drawings**

Fig. 1a is a flow chart of a method in accordance with an embodiment of the present invention.

Fig. 1b is a flow chart indicating a method of generating a voice service according to one embodiment of the present invention.

Fig. 1c is a flow chart indicating a method for interactive voice broadcasting according to an embodiment of the present invention.

Fig. 2 is a flow chart indicating a sequence of an interactive voice broadcast according to one embodiment of the present invention.

Fig. 3a is a schematic block diagram of a system in accordance with an embodiment of the present invention.

Fig. 3b is a schematic block diagram of an intelligence server according to an embodiment of the present invention.

Fig. 3c is a schematic block diagram of call server according to an embodiment of the present invention.

Fig. 4 is a schematic block diagram of a commercial transaction processing system according to an embodiment of the present invention.

Fig. 5 is a flow chart of a method of using a voice service bureau according to an embodiment of the present invention.

Fig. 6a is a schematic block diagram of a voice service system incorporating a voice service bureau according to one embodiment of the present invention.

Fig. 6b is block diagram of a primary voice bureau according to one embodiment of the present invention.

Fig. 6c is a block diagram of a backup voice bureau according to another embodiment of the present invention.

Fig. 7 is a flow chart illustrating a method for integrating inbound and outbound voice services.

Fig. 8 is a block diagram of a call server configured to provide integrated inbound and outbound voice services.

Fig. 9 illustrates a flowchart for processing of voice command input according to the invention.

Fig. 10 illustrates an architecture for voice command input according to the invention.

**Detailed Description of Preferred Embodiments**

In general, the invention generates and delivers interactive voice message services to one or more subscribers having a service and delivery profile, as described more fully below. The invention permits the management of a voice message session according to spoken commands and other input, using a voice input module 5004 and other elements to adaptively deliver menus and content, as depicted in Fig. 10.

A flowchart for voice command input according to the invention is shown in Fig. 9. In step 6000, processing begins. In step 6002, a voice service telecast to one or more of the subscribers of the service according to the invention is initiated and generated, according to steps more fully detailed below.

In step 6004, the initial voice service broadcast is delivered to a telephone or other two-way communication device for the subscriber. That broadcast may, for instance, contain or relate to financial, medical, news clipping, personal or other information desired by the subscriber. In step 6006, the recipient is presented with an authentication prompt, if configured for that subscriber's account. For instance, the call server 18 may query the user "Ms. Jones please say your PIN now" or another similar prompt. The subscriber's voice spectrum and other information may also be used to authenticate the recipient. Other authentication techniques, such as password or PIN entry by voice or keypad, may also be used in conjunction with the invention.

In step 6008, the call server may invoke voice input module 5004 to receive the voice input from the recipient of the voice service broadcast and convert the voice input to digital form. The voice input module 5004 may for instance contain and use voice digitizing and other

circuitry to sample the recipient's voice, convert the voice to digital values and process the digital values to determine cadence, gender, voice spectrum and other information. Commercially available speech detection packages, such as those by Nuance, Speechworks, Dragon and others, may likewise be incorporated or used. Speech detection engines may compare voice input to a prerecorded voice stamp to verify a speaker or identify speech. Other speech detection technology may be used.

In step 6010, the data generated by voice input module 5004 is passed to a discriminator module 5006 to determine the content of the recipient's voice input. Discriminator module 5006 may for instance incorporate or use natural language software, neural network or other pattern recognition modules, phoneme databases and other voice discrimination elements to identify units of communication as will be understood by persons skilled in the art.

In step 6012, the discriminator module 5006 determines the content of the recipient's voice input, such as "6072" in response to the PIN voice prompt. The call server 18 then processes the input so discriminated as in other embodiments described more fully below, to generate further information for delivery, receive further commands and complete the voice broadcast session. In step 6014, the recipient is authenticated according to the PIN or other information, and if validated processing proceeds to step 6018. If the recipient is not validated, control proceeds to step 6016 to test whether a predetermined number of attempts has been made. For example, a limit of three failed authentication attempts may be used. If that number is reached, control proceeds to step 6026 and processing ends. If not, control returns to step 6006 to reprompt the subscriber.

After successful authentication, in step 6018, the call server 18 presents the recipient with voice broadcast content and may present a further voice command prompt. That voice command prompt may be, for instance, "Ms. Jones would you like to sell Stock XYZ. Say Yes to sell, say No to decline" or similar.

Control proceeds to step 6020 in which the recipient's further voice input is received, converted and processed, and in step 6022 further information and/or voice commands are presented. In step 6024 termination of the voice broadcast session is tested for, for instance by voice command prompt to the recipient such as "Ms. Jones say 'Complete', if your broadcast is completed" or other. If the voice broadcast session is not complete, processing returns to step 6020. If the voice broadcast session is complete, control proceeds to step 6026 and processing ends.

An architecture for voice broadcast delivery and voice input according to the invention is shown in Fig. 10, which depicts an architecture generally similar to that of Fig. 3a. In this figure, a call server 18 acts to manage and deliver voice broadcast content to subscribers via a telephone interface 202, WWW interface 201 or otherwise. Call server 18 includes the voice input module 5004, which may include an analog to digital (A/D) converter chip such as those manufactured by Texas Instruments, Inc., Advanced Micro Devices and others, to sample the recipient's voice input in audio range over the public switched telephone network, or otherwise.

Voice input module in turn outputs a digital representation of the sampled voice data, for output to discrimination module 5006. Discrimination module 5006 may incorporate a neural network or other pattern recognition module to separate discreet voice commands or inputs from the sampled voice input, according to language models, vocabulary databases and other

components that will be appreciated by persons skilled in the art. Voice commands or inputs recognized by the invention may likewise include navigation commands to guide a subscriber backward, forward, to a specified menu page, main page or to another location in a menu or data sequence. Voice commands or inputs may likewise include responses to Yes/No prompts, numbered list prompts, option prompts (e.g. order red car, blue car) or other voice prompts such as "Which stock would you like a quote for" to which a subscriber may respond "Company X".

The voice menu presentation to the recipient may be controlled using administrator console 161 in conjunction with service wizard module 1616 or otherwise to create a series of information queries appropriate to the subscriber's account. The voice services for each voice command-enabled subscriber may be arranged and updated using intelligence backend server 163, which may for instance sort and select content for delivery according to the input identified by discriminator module 5006. The delivery of that content proceeds in a manner similar to other embodiments described herein.

According to an embodiment of the present invention in another regard, a system is provided for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers. For example, subscribers may be called by the system, and have content delivered audibly over the telephone or other voice-enabled terminal device. During the IVB, information may be exchanged between the system and a subscriber. The system conveys content to the subscriber and, the subscriber may respond by pressing one or more buttons on a telephone touch pad dial (or other input mechanism) to hear more information, to exercise options, or to provide other responses. This interaction shapes the structure of a basic exchange between the system

and the subscriber. During or after the call is terminated, the subscriber's responses may be stored and processed (*e.g.*, by other applications).

According to one embodiment of the present invention, a method for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers is provided. Figure 1a depicts a flow chart of a method for automatic, interactive, real-time, voice transmission of OLAP output according to one embodiment. The method begins in step 110 with the creation of a voice service (*e.g.*, by a system administrator or user). A voice service is created using, for example, a voice service wizard which may comprise a series of interfaces. One embodiment of a method for creating a voice service is explained in more detail below in conjunction with Figure 1b. One embodiment of a voice service wizard is explained below in conjunction with Figure 3b.

After a voice service is created, users may subscribe or be subscribed to the voice service (step 120), for example, by using a subscription interface. According to one embodiment, users may subscribe to an existing voice service over the telephone or by web-based subscription. A user may also be subscribed programmatically. In other embodiments, a user may subscribe to a voice service via electronic mail. Not every voice service created in step 110 is available for subscription. More specifically, according to another embodiment, only a user with appropriate access, such as the creator of the service, is allowed to subscribe himself or others to a service. Such a security feature may be set when the voice service is created.

In step 130, a scheduling condition or other predetermined condition for the voice services is monitored to determine when they are to be executed. That is, when a voice service is created or subscribed to, the creator or user specifies when the voice service is to be executed. A

user may schedule a voice service to execute according to the date, the time of day, the day of the week, etc. and thus, the scheduling condition will be a date, a time, or a day of the week, either one time or on a recurring basis. In the case of an alert service, discussed in more detail below, the scheduling condition will depend on satisfaction of one or more conditions. According to one embodiment, the condition(s) to be satisfied is an additional scheduling condition. According to another embodiment, to another embodiment, a service may be executed "on command" either through an administrator or programmatically through an API. Scheduling of voice services is discussed in more detail below.

The method continues monitoring the scheduling condition for voice services until a scheduling condition is met. When a scheduling condition is met, that voice service is executed. The execution of a voice service involves, inter alia, generating the content for the voice service, and structuring the voice service to be telecast through a call server. The execution of a voice service is explained in detail in conjunction with Figure 1c.

An example of a telecast is as follows.

PERSONALIZED GREETING

Hello Joe, this is your stock update.

PIN VERIFICATION

Please enter your six digit PIN number

(Joe enters his PIN, using the keypad dial on his telephone.)

MENU OPTIONS

Your portfolio was up by $1000 today.

Please select:

1. To get a portfolio stock update

2. To conduct a transaction

(Joe presses 2)

SUB MENU

Thank you, Joe! Please select a ticker.

1. PQT

2. TQP

3. Listen to options again

4. Return to main menu

(Joe presses 1.)

SUB MENU

Would you like to buy or sell stock? Please press:

1. To sell stock

2. To buy stock

(Joe presses 1.)

SUB MENU

How many shares of PQT would you like to sell today? Please press:

1. To sell 50 shares

2. To sell 100 shares

3. To sell 200 shares

4. To sell another quantity

(Joe presses 2.)

SUB MENU

You selected 2. You want to sell 100 shares of PQT. Please press:

1. If this is correct

2. If this is incorrect

3. If you want to change the number of shares you want to buy.

(Joe presses 1.)

END VOICE SERVICE/TERMINATE TELECAST

Thank you for using our voice interactive broadcasting service, Joe. We will call you

back when your transaction is completed. Good-bye.

As can be seen from the above sample during an IVB, the user is presented with information, *e.g.*, the status of his portfolio, and is presented options related to that report, *e.g.*, the option to buy or sell stock.

According to one embodiment, a voice service is constructed using service wizard. A voice service is constructed using several basic building blocks, or elements, to organize the content and structure of a call. According to one embodiment, the building blocks of a voice service comprise elements of a markup language. According to one particular embodiment, elements of a novel markup language based on XML (TML) are used to construct voice services. Before explaining how a telecast is constructed, it will be helpful to define these elements.

The DIALOG element is used to define a unit of interaction between the user and the system and it typically contains one or more of the other elements. A DIALOG can not be contained in another element.

The SPEECH element is used to define text to be read to a user.

The INPUT element is used to define a section of a DIALOG that contains interactive elements, *i.e.*, those elements that relate to a response expected from a user and its validation. An INPUT element may contain OPTION, PROMPT and ERROR elements.

An OPTION element identifies a predefined user selection that is associated with a particular input. According to one embodiment, OPTION elements are used to associate one or more choices available to a user with telephone keys.

A PROMPT element defines a particular input that is expected. According to one embodiment, a PROMPT element defines that a sequence or number of key presses from a telephone keypad is expected as input. Unlike an OPTION Element, a PROMPT Element is not associated with predefined user selections.

The PROMPT and OPTION elements may also be used to request user input using natural language. According to one embodiment, speech recognition technology is used to enable a user to respond to a PROMPT element or to select an OPTION element verbally by saying a number, e.g., “one.”. The verbal response is recognized and used just as a keypress would be used. According to another embodiment, the user may provide a free form verbal input. For example, a PROMPT element may request that a user enter, e.g., the name of a business. In response the user speaks the name of a business. That spoken name is then resolved against predetermined standards to arrive at the input. Word spotting and slot filling may also be used in conjunction with such a PROMPT to determine the user input. For example, a PROMPT may request that the user speak a date and time, e.g., to choose an airline flight or to make a restaurant reservation. The user’s spoken response may be resolved against known date and time formats to determine the input. According to another embodiment, a PROMPT is used to request input using natural language. For instance, in conjunction with a voice service to be used to make travel plans, instead of having separate PROMPT elements request input for flight arrival, departure dates and locations, a single natural language PROMPT may ask, “Please state your

travel plan." In response, the user states 'I'd like to go from Washington DC to New York city on the 3rd of January and return on the 3rd of February. This request would be processed using speech recognition and pattern matching technology to derive the user's input.

The ERROR element is used to define the behavior of the system if a user makes an invalid response such as touching a number that has not been associated with an OPTION element, or entering input that does not meet the criteria of a PROMPT element. A SYS-ERROR element defines a handler for certain events, such as expiration of the waiting time for a user response.

The FOR-EACH element is used to direct the system to loop through a list of variables *e.g.*, variables contained in a database report, or variables from a user input, to dynamically generate speech from data.

In addition to the elements described above, there are two features that maximize an administrator's ability to design voice services. Call Flow Reports enable an administrator to generate the structure of a call based on the content of an report *e.g.*, from an OLAP system or other data repository. For example, the options presented to a user in a PROMPT element may be made to correspond to the row of a data report. According to one embodiment, report data is converted into options by application of an XSL (extensible style sheet language) style sheet. The result of this application is inserted into the static call structure when the voice service is executed.

The use of an XSL style sheet is a feature that maximizes an administrator's voice service building ability. As discussed above, they are used to create dynamic call structure that depends

on data report output. They may also be used to generate a text string that comprises the message to be read to a user at any point in a call.

A method for creating a voice service according to one embodiment will now be explained in conjunction with Figure 2. The method begins in step 210 by naming the voice service. Then, in step 220 various scheduling parameters of the voice service are defined. In step 250 the service content is defined. And, in step 260, the personalization modes, or style properties are selected for the voice service.

According to one embodiment, in step 210, a voice service is named and a description of the voice service provided. By providing a name and description, a voice service may be uniquely identified. An interface is provided for prompting input of the name of the service to be created or edited. An input may also be provided for a written description. An open typing field would be one option for providing the description input. According to another embodiment, if an existing call service has been selected to edit, the service name field may not be present or may not allow modification.

In step 220, conditions for initiating the service are selected. This may include selecting and defining a service type. At least two types of services may be provided based on how the services are triggered. A first type of service is run according to a predetermined schedule and output is generated each time the service is run. A second type of service, an alert service, is one that is run periodically as well, however, output is only generated when certain criteria is satisfied. Other service types may be possible as well. In one embodiment the administrator is prompted to choose between a scheduled service or an alert service. An interface may provide an

appropriate prompt and some means for selecting between a scheduled service and an alert service. One option for providing the input might be an interface with a two element toggle list.

In one embodiment, a set of alert conditions is specified to allow the system to evaluate when the service should be initiated if an alert type service has been selected. In one embodiment, a report or a template/filter combination upon which the alert is based is specified. Reports and template/filter combinations may be predefined by other objects in the system including an agent module or object creation module. According to one embodiment, an agent module, such as DSS agent™ offered by MicroStrategy, may be used to create and define reports with filters and template combinations, and to establish the alert criteria for an alert service. According to another embodiment, an interface is be provided which includes a listing of any alert conditions presently selected for the voice service. According to this embodiment, the interface may comprise a display window. A browse feature may take the user to a special browsing interface configured to select a report or filter-template combination. One embodiment of an interface for selecting reports and filter-template combinations is described below. Once a report or filter and template combination is chosen, the alerts contained in the report or filter and template combination may be listed in the display window of the interface.

In step 220, the schedule for the service is also selected. According to one embodiment, predefined schedules for voice services may be provided or a customized schedule for the voice service may be created. If a new schedule is to be created, a module may be opened to enable the schedule name and parameters to be set. Schedules may be run on a several-minute, hourly, daily, monthly, semi-annual, annual or other bases, depending upon what frequency is desired. According to one embodiment, an interface is provided that allows the administrator to browse

through existing schedules and select an appropriate one. The interface may provide a browsing window for finding existing schedule files and a "new schedule" feature which initiates the schedule generating module. In one embodiment, schedules may not be set for alert type services. However, in some embodiments, a schedule for evaluating whether alert conditions have been met may be established in a similar manner.

In step 220, the duration of the service is also set. Service duration indicates the starting and stopping dates for the service. Setting a service duration may be appropriate regardless of whether a scheduled service or alert type service has been selected. The start date is the base line for the scheduled calculation, while the end date indicates when the voice service will no longer be sent. The service may start immediately or at some later time. According to one embodiment, interface is provided to allow the administrator to input start and end dates. The interface may also allow the administrator to indicate that the service should start immediately or run indefinitely. Various calendar features may be provided to facilitate selection of start and stop dates. For example, a calendar that specifies a date with pull-down menus that allow selection of a day, month and year may be provided according to known methods of selecting dates in such programs as electronic calendar programs and scheduling programs used in other software products. One specific aid that may be provided is to provide a calendar with a red circle indicating the present date and a blue ellipse around the current numerical date in each subsequent month to more easily allow the user to identify monthly intervals. Other methods may also be used.

In step 220, a voice service may also be designated as a mid-tier slicing service. In one embodiment, mid-tier slicing services generate content and a dynamic subscriber list in a single

query to an OLAP system. According to one embodiment, in a mid-tier slicing service a single database query is performed for all subscribers to the service. The result set developed by that query is organized in a table that contains a column that indicates one or more users that each row of data is applicable to.

In step 250, the content of the voice service is defined. Defining the content of the voice service may include selecting the speech to be delivered during the voice service broadcast (content), the structure of dialogs, menus, inputs, and the background procedures which generate both content and structure. In one embodiment, defining voice service content establishes the procedures performed by the vss server to assemble one or more active voice pages in response to initiation of the voice service. According to one embodiment, defining service content involves establishing a hierarchical structure of TML elements which define the structure and content of a voice service. All of the elements in a given service may be contained within a container.

The personalization type is selected in step 260. Personalization type defines the options that the administrator will have in applying personalization filters to a voice service. According to one embodiment, a personalization filter is a set of style properties that can be used to determine what content generated by the service will be delivered to the individual user and in what format it will be delivered. In one embodiment, personalizing the delivery format may include selection of style properties that determine the sex of the voice, the speed of the voice, the number of call back attempts, etc. Personalization filters may exist for individual users, groups of users, or types of users. According to one embodiment, personalization filters may be created independent of the voice service. According to this embodiment, a voice service

specifies what filters are used when generating IVBs. Some personalization type options may include: allowing no personalization filters; allowing personalization filters for some users, but not requiring them; and requiring personalization filters for all interactive voice broadcasts made using the service.

According to one embodiment, specifying personalization type is accomplished by administrator input through an interface. The interface may offer a toggle list with the three options: required personalization, optional personalization, and no personalization.

The voice service may be stored in a database structure to enable users to retrieve predefined voice services and to subscribe to these services, for example, through subscription interfaces explained in conjunction Figures 3a-3c or otherwise. An interface informing the administrator that creation of the voice service is complete may also be provided.

According to one embodiment, the method of Figure 1b also comprises an error condition step. An error condition step may be used to enable administrators to specify "error" conditions and the handling of those conditions. For example, an "error" condition may comprise a notification that a server is "down" or that there is no data to be returned. An administrator may specify particular actions to be performed by the system in response to one or more error conditions. For example, an administrator may specify an "addressing" error (*e.g.*, disconnected number) and indicate a particular action to be performed in response to an "addressing" error (*e.g.*, notify system administrator). Other error conditions might include: an alert report encountering an error and returning no data; a subscriber lacking the required personalization filter for the service; errors occurring in the generation of one or more reports; or reports returning no data. Various other conditions and actions may be specified. Certain error

conditions may be predetermined for the system, but an administrator may have reasons for supplementing or diverging from the predetermined error conditions. According to one particular embodiment, error conditions are specified using the ERROR and SYS-ERROR elements.

In one embodiment, setting error conditions may be accomplished using an error handling interface. The interface may allow the administrator to select either default error handling, or to customize error handling using a module for defining error handling. If default handling is selected, the system uses established settings. If customized handling is chosen, the user may use a feature to access the appropriate interface for the error handling module.

Servers may have limited capacity to perform all of the actions required of them simultaneously, the method of Figure 1b comprises a step for prioritizing the execution and delivery of voice services. Prioritization may establish the order in which the voice service system allocates resources for processing voice service and delivering the IVB. According to one embodiment, assigning priority to a voice service establishes priority for queries to the database system, formatting the voice service, or IVBs. Any criteria may be used for establishing priority. According to one embodiment, priority is established based on service content. According to another embodiment, priority is based on service destination. According to another embodiment, priority may be established based on the type of voice service, *i.e.*, alert vs. scheduled. Any number of procedures or criteria for denoting relative importance of service delivery may be established.

In one embodiment, an interface is provided for defining the priority of the voice service being created or edited. According to one embodiment, the interface comprises a screen

including option boxes with pull down menus listing the number of different prioritization options.

Another aspect of the invention relates to a method for executing a voice service. Figure 1c depicts one example of a flow chart for executing a voice service. In step 310, the content of a voice service is generated. In step 320, the call structure of a telecast is created via Active Voice Pages. In step 330, the AVPs are put in a call database for processing *e.g.*, in a call queue. In step 340, the call request is processed and an interactive voice broadcast with the user is implemented. In step 350, user's responses are written back to the voice service system (*e.g.*, the Active Voice Page). Each of these steps will be explained in more detail below.

According to one embodiment, content is created in step 310 as follows. A voice service execution begins by running scheduled reports, queries or by taking other action to determine whether the service should be sent. The subscribers for the service are then resolved. Datasets are generated for each group of subscribers that has unique personalization criteria.

Call structure may be created (step 320) as follows. An AVP contains data at various hierarchical content levels (nodes) that can be either static text or dynamic content. Static text can be generated *e.g.*, by typing or by incorporating a text file. Dynamic content may be generated *e.g.*, by inserting data from a data report using a grid an/or an XSL stylesheet. Moreover, content is not limited to text based information. Other media, such as, sound files, may be incorporated into the AVP. The call data (for example, at a particular level) may be the text that is converted to speech and played when the recipient encounters the node.

According to another embodiment, call content may include "standard" active voice pages that are generated and inserted into a database or Web Server where the pages are

periodically refreshed. According to one particular embodiment, the active voice page that is generated for a user contains links to these standard active voice pages. The links may be followed using a process similar to web page links.

The call structure may comprise either a static structure that is defined in the voice service interfaces *e.g.*, by typing text into a text box and/or a dynamic structure generated by grid/XSL combinations. The dynamic structure is merged with static structure during the service execution. A single call structure is created for each group of users that have identical personalization properties across all projects because such a group will receive the same content.

After a call structure is generated, in step 330, it is sent to a call database *e.g.*, call database 1811 shown in Figure 3calong with the addresses and style properties of the users. The style properties govern the behavior of a call server 18 in various aspects of the dialog with a user. Call server 18 queries call database 1811 for current call requests and places new call requests in its queue.

In step 340, a call request is processed. A call is implemented on call server 18 using one of several ports that are configured to handle telephone communication. When a port becomes available, the call request is removed from the queue and the call is made to the user. As the user navigates through an active voice page, *e.g.*, by entering input using the key pad or by speaking responses, call/content is presented by converting text to speech in text-to-speech engine 1814. User input during the call may be stored for processing. According to another embodiment, user responses and other input may also be used to follow links to other active voice pages. For example, as explained above, "standard" active voice pages may be generated and inserted into a database or Web Server. Then, when a user's voice service is delivered, that voice service may

contain links to information that may be accessed by a user. A user may access those standard active voice pages by entering input in response to OPTION or PROMPT elements.

In step 350, user responses are stored by the system. According to one embodiment, user responses are stored in a response collection defined by the active voice page. A voice service may specify that a subscriber return information during an IVB so that another application may process the data. For instance, a user may be prompted to purchase a commodity and be asked to enter or speak the number of units for the transaction. During or after an IVB, the subscriber's responses are written to a location from which they can be retrieved for processing (*e.g.*, by an external application).

Fig. 2 is an example of an IVB with interactive call flow. An IVB usually contains a greeting message that addresses the targeted user, identifies the name of the calling application, and states the purpose of the call and/or presents summary metrics. The voice service system can also implement a PIN verification protocol, if this layer of security is required. The main menu structure of an IVB can contain a number of options that lead to sub-menu structures. A menu can also contain prompts for the user to enter numerical information using a telephone touch pad dial. A node along the hierarchical menu structure may have options to return the user to a higher level.

Fig. 3 depicts an embodiment of a system according to one embodiment of the present invention. Preferably, the system comprises database system 12, a DSS server 14, voice service server 16, a call server 18, subscription interface 20, and other input/files 24.

Database system 12 and DSS server 14 comprise an OLAP system that generates user-specified reports from data maintained by database system 12. Database system 12 may comprise any data warehouse or data mart as is known in the art, including a relational database

management system ("RDBMS"), a multidimensional database management system ("MDDBMS") or a hybrid system. DSS server 14 may comprise an OLAP server system for accessing and managing data stored in database system 12. DSS server 14 may comprise a ROLAP engine, MOLAP engine or a HOLAP engine according to different embodiments. Specifically, DSS server 14 may comprise a multithreaded server for performing analysis directly against database system 12. According to one embodiment, DSS server 14 comprises a ROLAP engine known as DSS Server™ offered by MicroStrategy.

Voice service server (VSS) 16, call server 18 and subscription interface 20 comprise a system through which subscribers request data and reports *e.g.*, OLAP reports through a variety of ways and are verbally provided with their results through an IVB. During an IVB, subscribers receive their requested information and may make follow-up requests and receive responses in real-time as described above. Although the system is shown, and will be explained, as being comprised of separate components and modules, it should be understood that the components and modules may be combined or further separated. Various functions and features may be combined or separated

Subscription interface 20 enables users or administrators of the system to monitor and update subscriptions to various services provided through VSS 16. Subscription interface 20 includes a world wide web (WWW) interface 201, a telephone interface 202, other interfaces as desired and a subscriber API 203. WWW interface 201 and telephone interface 202 enable system 100 to be accessed, for example, to subscribe to voice services or to modify existing voice services. Other interfaces may be used. Subscriber API 203 provides communication

between subscription interface 20 and VSS 16 so that information entered through subscription interface 20 is passed through to VSS 16.

Subscription interface 20 is also used to create a subscriber list by adding one or more subscribers to a service. Users or system administrators having access to VSS 16 may add multiple types of subscribers to a service such as a subscriber from either a static recipient list (SRL) (*e.g.*, addresses and groups) or a dynamic recipient list (DRL) (described in further detail below). The subscribers may be identified, for example, individually, in groups, or as dynamic subscribers in a DRL. Subscription interface 20 permits a user to specify particular criteria (*e.g.*, filters, metrics, etc.) by accessing database system 12 and providing the user with a list of available filters, metrics, etc. The user may then select the criteria desired to be used for the service. Metadata may be used to increase the efficiency of the system.

A SRL is a list of manually entered names of subscribers of a particular service. The list may be entered using subscription interface 20 or administrator console 161. SRL entries may be personalized such that for any service, a personalization filter (other than a default filter) may be specified. A SRL enables different personalizations to apply for a login alias as well. For example, a login alias may be created using personalization engine 1632. Personalization engine 1632 enables subscribers to set preferred formats, arrangements, etc. for receiving content. The login alias may be used to determine a subscriber's preferences and generate service content according to the subscriber's preferences when generating service content for a particular subscriber.

A DRL may be a report which returns lists of valid user names based on predetermined criteria that are applied to the contents of a database such as database system 12. Providing a

DRL as a report enables the DRL to incorporate any filtering criteria desired, thereby allowing a list of subscribers to be derived by an application of a filter to the data in database system 12. In this manner, subscribers of a service may be altered simply by changing the filter criteria so that different user names are returned for the DRL. Similarly, subscription lists may be changed by manipulating the filter without requiring interaction with administrator console 161. Additionally, categorization of each subscriber may be performed in numerous ways. For example, subscribers may be grouped via agent filters. In one specific embodiment, a DRL is created using DSS Agent™ offered by MicroStrategy.

VSS 16 is shown in more detail in Figure 3b. According to one embodiment, VSS 16 comprises administrator console 161, voice service API 162 and backend server 163. Administrator console 161 is the main interface of system 100 and is used to view and organize objects used for voice broadcasting. Administrator console 161 provides access to a hierarchy of additional interfaces through which a system administrator can utilize and maintain system 100. Administrator console 161 comprises system administrator module 1611, scheduling module 1612, exceptions module 1613, call settings module 1614, address handling module 1615, and service wizard 1616.

System administrator module 1611 comprises a number of interfaces that enable selection and control of the parameters of system 100. For example, system administrator module 1611 enables an administrator to specify and/or modify an email system, supporting servers and a repository server with which system 100 is to be used. System administrator 1611 also enables overall control of system 100. For example, system administrator module is also used to control

the installation process and to start, stop or idle system 100. According to one embodiment, system administrator 1611 comprises one or more graphical user interfaces (GUIs).

Scheduling module 1612 comprises a number of interfaces that enable scheduling of voice services. Voice services may be scheduled according to any suitable methodology, such as according to scheduled times or when a predetermined condition is met. For example, the predetermined condition may be a scheduled event (time-based) including, day, date and/or time, or if certain conditions are met. In any event, when a predetermined condition is met for a given service, system 100 automatically initiates a call to the subscribers of that service. According to one embodiment, scheduling module 1612 comprises one or more GUIs.

Exceptions module 1613 comprises one or more interfaces that enable the system administrator to define one or more exceptions, triggers or other conditions. According to one embodiment, exceptions module 1613 comprises one or more GUIs.

Call settings module 1614 comprises one or more interfaces that enable the system administrator to select a set of style properties for a particular user or group of users. Each particular user may have different options for delivery of voice services depending on the hardware over which their voice services are to be delivered and depending on their own preferences. As an example of how the delivery of voice services depends on a user's hardware, the system may deliver voice services differently depending on whether the user 's terminal device has voice mail or not. As an example of how the delivery of voice services depends on a user's preferences, a user may chose to have the pitch of the voice, the speed of the voice or the sex of the voice varied depending on their personal preferences. According to one embodiment, call settings module 1614 comprises one or more GUIs.

Address handling module 1615 comprises one or more interface that enable a system administrator to control the address (*e.g.*, the telephone number) where voice services content is to be delivered. The may be set by the system administrator using address handling module 1615. According to one embodiment, address handling module 1615 comprises one or more GUIs.

Voice service wizard module 1616 comprises a collection of interfaces that enable a system administrator to create and/or modify voice services. According to one embodiment, service wizard module 1616 comprises a collection of interfaces that enable a system administrator to define a series of dialogs that contain messages and inputs and determine the call flow between these dialogs based on selections made by the user. The arrangement of the messages and prompts and the flow between them comprises the structure of a voice service. The substance of the messages and prompts is the content of a voice service. The structure and content are defined using service wizard module 1616.

Voice service API 162 (*e.g.*, MicroStrategy Telecaster Server API) provides communication between administrator console 161 and backend server 163. Voice Service API 162 thus enables information entered through administrator console 161 to be accessed by backend server 163 (*e.g.*, MicroStrategy Telecaster Server).

Backend server 163 utilizes the information input through administrator console 161 to initiate and construct voice services for delivery to a user. Backend server 163 comprises report formatter 1631, personalization engine 1632, scheduler 1633 and SQL engine 1634. According to one embodiment, backend server 163 comprises MicroStrategy Broadcast Server. Report formatter 1631, personalization engine 1632, and scheduler 1633 operate together, utilizing the

parameters entered through administrator console 161, to initiate and assemble voice services for transmission through call server 18. Specifically, scheduler 1633 monitors the voice service schedules and initiates voice services at the appropriate time. Personalization engine 1632 and report formatter 1631 use information entered through service wizard 1616, exceptions module 1613, call settings module 1614, and address module 1615, and output provided by DSS server 14 to assemble and address personalized reports that can be sent to call server 18 for transmission. According to one embodiment, report formatter 1631 includes an XML based markup language engine to assemble the voice services. In a particular embodiment, report formatter includes a Telecaster Markup Language engine offered by MicroStrategy Inc. to assemble the call content and structure for call server 18.

SQL engine 1634 is used to make queries against a database when generating reports. More specifically, SQL engine 1634 converts requests for information into SQL statements to query a database.

Repository 164 may be a group of relational tables stored in a database. Repository 164 stores objects which are needed by system 100 to function correctly. More than one repository can exist, but preferably the system 100 is connected to only one repository at a time.

According to one embodiment, a call server 18 is used to accomplish transmission of the voice services over standard telephone lines. Call server 18 is shown in more detail in Figure 3c. According to one embodiment, call server 18 comprises software components 181 and hardware components 182. Software components 181 comprise call database 1811, mark-up language parsing engine 1812, call builder 1813, text-to-speech engine 1814, response storage device 1815 and statistic accumulator 1816.

Call database 1811 comprises storage for voice services that have been assembled in VSS 16 and are awaiting transmission by call server 18. These voice services may include those awaiting an initial attempt at transmission and those that were unsuccessfully transmitted (*e.g.*, because of a busy signal) and are awaiting re-transmission. According to one embodiment, call database 1811 comprises any type of relational database having the size sufficient to store an outgoing voice service queue depending on the application. Call database 1811 also comprises storage space for a log of calls that have been completed.

Voice services stored in call database 1811 are preferably stored in a mark-up language. Mark-up language parsing engine 1812 accepts these stored voice services and separates the voice services into parts. That is, the mark-up language version of these voice services comprises call content elements, call structure elements and mark-up language instructions. Mark-up language parsing engine 1812 extracts the content and structure from the mark-up language and passes them to call builder 1813.

Call builder 1813 is the module that initiates and conducts the telephone call to a user. More specifically, call builder dials and establishes a connection with a user and passes user input through to markup language parsing engine 1812. In one embodiment, call builder 1813 comprises "Call Builder" software available from Call Technologies Inc. Call builder 1813 may be used for device detection, line monitoring for user input, call session management, potentially transfer of call to another line, termination of a call, and other functions.

Text-to-speech engine 1814 works in conjunction with mark-up language parsing engine 1812 and call builder 1813 to provide verbal communication with a user. Specifically, after call

builder 1813 establishes a connection with a user, text-to-speech engine 1814 dynamically converts the content from mark-up language parsing engine 1812 to speech in real time.

A voice recognition module may be used to provide voice recognition functionality for call server 181. Voice recognition functionality may be used to identify the user at the beginning of a call to help ensure that voice services are not presented to an unauthorized user or to identify if a human or machine answers the call. This module may be a part of call builder 1813. This module may also be used to recognize spoken input (say "one" instead of press "1"), enhanced command execution (user could say "transfer money from my checking to savings"), enhanced filtering (instead of typing stock symbols, a user would say "MSTR"), enhanced prompting, (saying numeral values).

User response module 1815 comprises a module that stores user responses and passes them back to intelligence server 16. Preferably, this is done within an AVP. During a telephone call, a user may be prompted to make choices in response to prompts by the system. Depending on the nature of the call, these responses may comprise, for example, instructions to buy or sell stock, to replenish inventory, or to buy or rebook an airline flight. User response module 1815 comprises a database to store these responses along with an identification of the call in which they were given. The identification of the call in which they were given is important to determining what should be done with these responses after the call is terminated. User responses may be passed back to intelligence server 16 after the call is complete. The responses may be processed during or after the call, by the system or by being passed to another application.

Statistics accumulator 1816 comprises a module that accumulates statistics regarding calls placed by call builder 1813. These statistics including, for example, the number of times a particular call has been attempted, the number of times a particular call has resulted in voice mail, the number of times a user responds to a call and other statistics, can be used to modify future call attempts to a particular user or the structure of a voice service provided to a particular user. For example, according to one embodiment, statistics accumulator 1816 accumulates the number of times a call has been unsuccessfully attempted by call builder 1813. This type of information is then used by call server 18 to determine whether or not the call should be attempted again, and whether or not a voice mail should be left.

Call server 18 also comprises certain hardware components 182. As shown in Figure 1c, hardware components 182 comprise processor 1821 and computer telephone module 1822. According to one embodiment, processor 1821 comprises a Pentium II processor, available from Intel, Inc. Module 1822 provides voice synthesis functionality that is used in conjunction with Text to Speech engine 1814 to communicate the content of voice services to a user. Module 1822 preferably comprises voice boards available from Dialogic, Inc. Other processors and voice synthesizers meeting system requirements may be used.

The system and method of the present invention may form an integral part of an overall commercial transaction processing system.

According to one embodiment of the present invention, a system and method that enable closed-loop transaction processing are provided. The method begins with the deployment of an IVB by executing a service. As detailed above, this includes generating the content and combining this with personalization information to create an active voice page. Call server 18

places a call to the user. During the call, information is delivered to the user through a voice-enabled terminal device (e.g., a telephone or cellular phone).

During the IVB, a user may request a transaction, service, further information from the database or other request, *e.g.*, based on options presented to the user. These will generically be referred to as transactions. The request may be, but is not necessarily, based on or related to information that was delivered to the user. According to one embodiment, the request comprises a user response to a set of options and/or input of information through a telephone keypad, voice input or other input mechanism. According to another embodiment, the request can be made by a user by speaking the request. Other types of requests are possible.

According to one embodiment, the user responses are written to a response collection, which along with information stored in the active voice page, can be used to cause a selected transaction to be executed. According to one embodiment, the active voice page comprises an XML-based document that includes embedded, generic requests, *e.g.*, a request for a transaction, or a request for additional information (a database query). These embedded requests are linked with, for example option statements or prompts so that when a user enters information, the information is entered into the generic request and thus completes a specific transaction request. For example, in the example if a user exercises an option to buy a particular stock, that stock's ticker symbol is used to complete a generic "stock buy" that was embedded in the active voice page.

According to one embodiment, tokens are used to manage user inputs during the IVB. A token is a temporary variable that can hold different values during an IVB. When a user enters input, it is stored as a token. The token value is used to complete a transaction request as

described above. According to one embodiment, the system maintains a running list of tokens, or a response collection, during an IVB.

In order to complete the requested transaction, the user responses (and other information from the active voice page) may need to be converted to a particular format. The format will depend, for example, on the nature and type of transaction requested and the system or application that will execute the transaction. For example, a request to purchase goods through a web-site may require the information to be in HTML/HTTP format. A request for additional information may require and SQL statement. A telephone-based transaction may require another format.

Therefore, the transaction request is formatted. According to one embodiment, the transaction is formatted to be made against a web-based transaction system. According to another embodiment, the transaction request is formatted to be made against a database. According to another embodiment, the transaction is formatted to be made against a telephone-based transaction system. According to another embodiment, the transaction is formatted to be made via e-mail or EDI. Other embodiments are possible.

In one embodiment, the formatted transaction request comprises an embedded transaction request. The system described in connection with Figures 1-3 provides interactive voice services using TML, a markup language based on XML. Using TML active voice pages are constructed that contain the structure and content for a interactive voice broadcast including, inter alia, presenting the user with options and prompting the user for information. Moreover in connection with OPTION and PROMPT elements, active voice pages also can include embedded statements

such as transaction requests. Therefore, the formatting for the transaction request can be accomplished ahead of time based on the particular types of transactions the user may select.

For example, in connection with an exemplary stock purchase, an active voice page can include an embedded transaction request to sell stock in the format necessary for a particular preferred brokerage. The embedded statement would include predefined variables for the name of the stock, the number of shares, the type of order (market or limit, etc.), and other variables. When the user chooses to exercise the option to buy or sell stock, the predefined variables are replaced with information entered by the user in response to OPTION or PROMPT elements. Thus, a properly formatted transaction request is completed.

In the system of Figures 1-3, TML parsing engine in call server 18 includes the functionality necessary to generate the properly formatted transaction request as described above. For example, in connection with the embodiment described above, the TML parsing engine shown in Figure 3c reads the active voice pages. When the TML parsing engine reads an OPTION element that includes and embedded transaction request, it stores the transaction request, and defines the necessary variables and variable locations. When the user exercises that OPTION, the user's input is received by the TML parsing engine and placed at the memory locations to complete the transaction request This technique could be used, for example, to generate a formatted transaction request for web-site.

According to another embodiment, where the transaction request is made via a natural language, voice request, a formatted transaction request can be generated in a number of ways. According to one embodiment, speech recognition technology is used to translate the user's request into text and parse out the response information. The text is then used to complete an

embedded transaction request as described above. According to another embodiment, speech recognition software is used to translate the request to text. The text is then converted to a formatted request based on a set of known preferences.

A connection is established with the transaction processing system. This can be accomplished during, or after the IVB. According to one embodiment, the transaction processing system comprises a remotely located telephone-based transaction site. For example, in the system shown in Figures 1-3, call server 18, through the TML parsing engine 1812, establishes a connection with a telephone-based transaction processing site.

According to another embodiment, the transaction processing system comprises a remotely based web-site. According to this embodiment, the formatted request includes a URL to locate the web-site and the system accesses the site through a web connection using the formatted request. Alternatively, the formatted request includes an e-mail address and the system uses any known email program to generate an e-mail request for the transaction.

After the connection is established, the transaction is processed by the transaction processing site and the user is notified of the status of the transaction. If the transaction is completed in real-time, the user may be immediately notified. If the transaction is executed after the IVB, the user may be called again by the system, sent an e-mail, or otherwise notified when the transaction has been completed.

According to one particular embodiment, the system comprises the interactive voice broadcasting system shown and described in Figures 1-3 and the transaction is accomplished in real-time. In this embodiment, confirmation of the transaction is returned to TML parsing engine 1812 shown in Figure 3 and translated to speech in text-to-speech engine 1814 and presented to

the user during the IVB. More specifically, and similar to the process described with respect to embedded formatted transaction requests, TML also enables embedding of a response statement. Thus, when the transaction is processed and confirmation of the transaction is returned to the system, an embedded confirmation statement is conveyed to the user through TML parsing engine 1812 after being converted to speech in text-to-speech engine 1814.

Figure 4 schematically depicts one example of how the system and method of the present invention would fit into such a commercial transaction processing system. Working from left to right in Figure 4, the system begins and ends with information stored in relational databases. One of the primary purposes of information is in making decisions. Thus, the information in the databases is most useful if provided to someone who desires it in a timely fashion.

A voice service system is provided to enable access to the information in the databases. The voice service system utilizes personalization information and personalized menus to construct AVPs pages that enable the information to be delivered to a user verbally. Moreover, the AVPs pages, not only enable information to be presented to the user. But, they also enable the user to provide information back to the voice service system for additional processing.

According to the embodiment shown in Figure 4, once the AVPs are constructed by voice service system, they are processed and the content is delivered to a user verbally in an IVB. Thus, call processing and text-to-speech technology are used to establish a telephone connection with a user and convert the active voice pages to speech for presentation to the user. As shown in Figure 4, the IVB may be delivered to a user in many devices, including a telephone, a mobile phone, voice mail, an answering machine or any other voice-enabled device.

During the IVB, depending on the content that is being delivered, control may be passed to an e-commerce application for the user to complete a transaction based on the information presented. For example, if the user has requested information about sales on a particular brand of merchandise, the user may be connected with a particular retailer in order to complete a transaction to buy a particular good or service. Information about this transaction is then added to the databases and thus may be advantageously accessed by other users.

It may not be economical for some potential users of a voice broadcasting system to buy and/or maintain their own telephony hardware and software as embodied in call server 18. In such a case, a voice service bureau may be maintained at a remote location to service users voice service requests. A voice service bureau and a method of using a voice service bureau according to various embodiments of the present invention is described in conjunction with Figures 5-6.

In one embodiment, a voice service bureau may comprise one or more call servers and call databases that are centrally located and enable other voice service systems to generate a call request and pass the call request to the VSB to execute a call. In this way the other voice service systems do not need to invest in acquiring and maintaining call data bases, call servers, additional telephone lines and other equipment or software. Moreover, the VSB facilitates weeding out usage of illegal numbers and spamming by number checking implemented through its web server.

A voice service bureau and a method of using a voice service bureau according to one embodiment are described in conjunction with Figures 5-6. Figure 5 depicts a method of utilizing a voice service bureau according to one embodiment of the present invention. The

method begins in step 810 with a request to place one or more telephone calls received through a computer network.

According to one embodiment, the voice service bureau is maintained at a location distant from the voice service system. Therefore, in order for a voice service to be processed by the voice service bureau, in step 810 the voice service is sent to the voice services bureau, preferably over some secure line of communication. According to one embodiment, the request is sent to the voice service bureau through the Internet using secure HTTPS. HTTPS provides a secure exchange of data between clients and the voice service bureau using asymmetric encryption keys based on secure server certificates. In another embodiment, SSL HTTP protocol is used to send a call request to the voice service bureau. Both of these protocols help ensure that a secure channel of communication is maintained between the voice service system and the voice service bureau. Other security techniques may be used.

When a request for a call or telecast is received, by the VSB, the request is authenticated by the voice service bureau in step 820. According to one embodiment, the authenticity of the request is determined in at least two ways. First, it is determined whether or not the request was submitted from a server having a valid, active server certificate. More specifically, requests may be typically received via a stream of HTTPS data. Each such request originating from a server with a valid server certificate will include an embedded code (i.e., server certificate) that indicates the request is authentic. In addition to the use of server certificates, each request may also be authenticated using an identification number and password. Therefore, if the request submitted does not include a valid server certificate and does not identify a valid I.D./password combination, the request will not be processed. The step of authenticating also comprises

performing any necessary decryption. According to one embodiment, any errors that are encountered in the process of decrypting or authenticating the call request are logged an error system and may be sent back to the administrator of the sending system. Other methods of authenticating the request are possible.

Each properly authenticated request is sent to a call server (step 830) and processed (step 840). According to one embodiment, the voice service bureau comprises a number of call servers. According to one embodiment, the calls are sent to a call database, and processed as set forth herein in conjunction with the explanation of call server 18.

One embodiment of a voice service bureau will now be explained in conjunction with Figures 6a-6c. Figure 6a depicts a system comprising a plurality of client side installations 91, a primary voice bureau 92, a system administrator 93, a backup voice service bureau 94, and a plurality of users 95. Client side installations 91 communicate with voice service bureau 92 through a computer network. Voice service bureau 92 communicates with users 95 through a voice network. According to one embodiment, the computer network comprises the internet and client side installations 91 communicate with voice service bureau 92 using HTTPS as described above, and the voice network comprises a public telephone network.

According to one embodiment, client side installations 91 are substantially identical to the system shown in Figure 4 except for the elimination of call server 18. In the system of Fig. 6a, the functionality of call server 18 is performed by VSB 92. As shown in this embodiment, VSB 92 can service multiple client side installations $91_1$ to 91n. According to another embodiment, client-side installation functionality may be included within VSB 92. According to

this embodiment VSB 92 constitutes a fully functional voice service that is accessible through email, telephone or other interfaces.

According to this embodiment, when voice services have been assembled by intelligence server 16, a request to have the voice services transmitted is sent via a secure network connection through the computer network shown to primary voice bureau 92 and backup voice service bureau 94 as described above. According to one embodiment, the request comprises a mark-up language string that contains the voice service structure and content and personal style properties and other information. As described above, voice bureau 92 authenticates the request, queues the voice services and sends telecasts to users 95 through the voice network.

A block diagram of one embodiment of primary voice bureau 92 is shown in Figure 6b. According to this embodiment, primary voice bureau comprises routers 921, dual-homed servers 922, database servers 923, call database 924, backup storage 925, call servers 926, internal switch 927, and system administrator 928. Routers 921 receive call requests via a computer network and pass them along to one of the two dual-homed servers 922. Router 921 monitors activity on servers 922 and forwards call requests to one of the two depending on availability.

Dual-homed servers 922 comprise servers configured to receive and send HTTPS email. As part of their receiving function, dual-homed servers 922 are configured to perform the authentication processing described above. According to one embodiment, dual-homed servers 922 determine whether the incoming request originated from a server with an active server certificate and also determine if the request contains a valid I.D./password combination. Once dual-homed servers 922 have authenticated the incoming request, they forward the request to be queued in call database 924. As part of their sending function, dual-homed servers 922 are

configured to format and send HTTPS email. As discussed above, during a telecast a user may request that further information be accessed from a database or that some transaction be performed. According to one embodiment, these user requests are forwarded back to the originating system via HTTPS email by dual-homed servers 922. Dual-homed servers 922 are load balanced to facilitate optimal performance and handling of incoming call requests.

Database servers 923, call database 924, and backup storage 925 together comprise a call request queuing system. Primary voice bureau 92 is configured to handle a large number of call requests. It may not be possible to process call requests as they arrive. Therefore, call requests are queued in call database 924. According to one embodiment, call database 924 comprises a relational database that maintains a queue of all call requests that need to be processed as well as logs of calls that have been processed. According to another embodiment, primary VSB 92 may include a failover measure that enables another system server to become the call database if call database 924 should fail.

Database servers 923 are configured to control access to call database 924. According to one embodiment, database servers may be optimized to generate SQL statements to access entries in call database at high speed. Database servers 923 also control storage of call requests and call logs in call database 924.

Call servers 926 each are configured to format and send telecasts. According to one embodiment, each of call servers 926 is substantially identical to call server 18 shown in Figure 3c. More specifically, each of call servers 926 receives requests for telecasts, parses the call content from the mark-language, establishes a connection with the user through phone lines 929,

and receives user responses. According to one embodiment, call servers 926 comprise a clustered architecture that facilitates message recovery in the event of server failure.

Primary voice bureau 92 is controlled by system administrator 93 and internal switch 927. System administrator controls switch 927 and thus controls the flow of call requests to call database 924 from dual homed servers 922 and to call servers 926 from call database 924.

System administrator 93 is also configured to perform a number of other services for primary voice bureau 92. According to one embodiment, system administrator 93 also comprises a billing module, a statistics module, a service module and a security module. The billing modules tabulates the number of voice service requests that come from a particular user and considers the billing plan that the customer uses so that the user may be appropriately billed for the use of voice bureau 92. The statistics module determines and maintains statistics about the number of call requests that are processed by voice bureau 92 and statistics regarding call completion such as, e.g., success, failed due to busy signal and failed due to invalid number. These statistics may be used, for example, to evaluate hardware requirements and modify pricing schemes. The security module monitors activity on voice bureau 92 to determine whether or not any unauthorized user has accessed or attempted to access the system. The service module provides an interface through which primary voice bureau 92 may be monitored, for example, to determine the status of call requests. Other service modules are possible. Moreover, although these services are described as distinct modules, their functionality could be combined and provided in a single module.

Backup voice service bureau 94 receives a redundant request for voice services. Backup voice service bureau 94 processes the requests only when primary voice service bureau is offline

or busy. One embodiment of backup voice service bureau 94 is shown in Figure 6c. Backup voice bureau 94 comprises routers 941, HTTP server 942, database server 943, call server 946 and routers 947. Each of these components performs a function identical to the corresponding element in primary voice bureau 92. Router 947 replaces switch 927. Router 947 controls the forwarding of call requests to database server 943 for queuing in an internal database, and the forwarding of call requests to call server 946 from database server 943.

The systems and methods discussed above are directed to outbound broadcasting of voice services. Nevertheless, in certain situations, for example when the out bound telecast is missed, it is desirable to for a voice service system to enable inbound calling. According to another embodiment, a method and system for providing integrated inbound and outbound voice services is disclosed.

A method for providing inbound access to voice services according to one embodiment of the present invention is shown in Figure 7. According to Figure 7, the method begins with receipt of a call requesting voice services in step 1210. To help ensure system integrity and to prevent unauthorized access, a call request is authenticated in step 1220. According to one embodiment, each incoming caller is automatically prompted to enter a login identifier and a PIN. According to another embodiment, an automatic number identification system is used, in addition to a login identifier and PIN system, to determine whether or not the user is calling from an authorized device. According to another embodiment, speaker recognition technology is utilized to identify a caller. According to this embodiment, voice prints for each user of the voice service system are stored as identifiers. When an inbound call is connected, pattern

matching techniques are used verify the user's speech against the previously stored voice prints. Other security measures are possible.

In step 1230, a voice page is located. As explained above, a telecast of a voice service is driven by an active voice page. Accordingly, a user calling in to access voice services locates the desired active voice page. According to one embodiment, the user is automatically placed into an active voice page of a voice service that the user missed. That is, the system chooses an active voice page that it was unable to deliver. According to this embodiment, when a call is undeliverable (e.g., when an answering machine picks up), the active voice page for that call is placed in memory in a "voice site" table or as an active voice page on a web site and addressed using the user's identification. When the user calls in to retrieve the voice service, after the user logs in, the table or web site will be searched for an active voice page that corresponds to their identification. If such a page exists, it is executed by the call server.

Other possibilities exist for accessing active voice pages through inbound calling. According to another embodiment, the system maintains a log of all voice services sent and provides an inbound user an option to select one of their previous voice services. According to another embodiment, an inbound caller is automatically placed into an active voice page that presents the user with an option to select one of that user's most frequently used services. According to still another embodiment, the user is allowed to search for past active voice pages by date or content. For example, the user may be prompted to enter a date on or near which the desired voice page was executed. According to another embodiment, the user may use the telephone keys to enter a search term and search the content of any previously executed active voice page that they are authorized to access or that is not secure.

Once an active voice page is located, the user navigates through the active voice page in step 1240. As described above, a user navigates through an active voice by exercising options, responding to prompts and otherwise entering input to the system. An inbound calling system would thus have access to the full functionality of the voice service system described in conjunction with Figures 1-6.

Figure 8 depicts a block diagram of a call server 18a that enables integrated inbound and outbound calling. In addition to the modules depicted in call server 18 of Figure 3, call server 18a comprises call receiver module 1817, security module 1818 and search module 1819. Moreover, in the system for permitting inbound and outbound calling, call database 1811 has been replaced with an enhanced call database 1811a.

In order to receive inbound calls, call server 18a comprises call receiver module 1817. Although, call server 18 discussed above contains hardware permitting reception of calls as well as transmission of calls, it is not set up to receive calls. Call receiver module 1817 enables call server 18a to receive calls and routes the incoming calls to security module 1818. According to one embodiment, call receiver module comprises a software component designed to configure call server 18a to receive calls. Other embodiments are possible.

Received calls are forwarded to security module 1818 for authentication. According to one embodiment discussed above, incoming calls are authenticated using login I.D.'s and passwords. According to another embodiment, automatic number identification software is used to identify and authenticate callers. According to another embodiment, speech recognition and pattern matching techniques are used to identify a caller.

Authenticated calls may search for an active voice page using search module 1819. According to one embodiment, search module 1819 comprises a search engine designed specifically to search active voice pages. According to one embodiment discussed above, active voice pages utilize an XML-based language and search module 1819 comprises an XML-based search engine. According to another embodiment, search module 1819 comprises a SQL engine designed to make queries against a relational or other type of database.

The active voice pages that are being search are stored in enhanced call database 1811a. In addition to its facilities to queue and log calls, enhanced call database 1811 includes facilities to catalog active voice pages. According to one embodiment, enhanced call database comprises a relational or other type of database. According to this embodiment, enhanced call database is used to store and categorize active voice pages and corresponding parameters, such as expiration dates for active voice pages. Other storage facilities are possible.

Various features and functions of the present invention extend the capabilities of previously known information delivery systems. One such system is MicroStrategy's Broadcaster version 5.6. The features and functions of the present invention are usable in conjunction with Broadcaster and other information delivery systems or alone. Other products may be used with the various features and functions of the invention including, but not limited to, MicroStrategy's known product suite.

Other embodiments and uses of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification and examples should be considered exemplary only. The scope of the invention is only limited by the claims appended hereto.

**What is claimed is:**

1. A system for the delivery of voice messages to a voice service subscriber using voice commands, comprising:

an input module, the input module sensing a voice input command from the subscriber; and

a content delivery module, communicating with the input module, the content delivery module selecting at least one of a plurality of voice messages to deliver according to the voice input command.

2. The system of claim 1, wherein the input module comprises an analog to digital converter which converts the voice input command to digital voice data.

3. The system of claim 2, wherein the input module stores the digital data.

4. The system of claim 1, further comprising a discriminator module, the discriminator module communicating with the input module and the content delivery module and identifying the digital voice data as at least one of a plurality of predetermined commands.

5. The system of claim 4, wherein the content delivery module presents the subscriber with voice message content according to the digital voice data.

6. The system of claim 5, wherein the content delivery module presents the subscriber with at least one voice command prompt to query voice input from the subscriber.

7. The system of claim 6, wherein the voice command prompt comprises a sequence of voice command prompts.

8. The system of claim 7, wherein the sequence of voice command prompts comprises a set of voice command prompts adaptively presented according to the digital voice data.

9. The system of claim 1, wherein the input module authenticates the subscriber for receipt of the voice messages.

10. The system of claim 9, wherein the authentication comprises at least one of PIN verification and voice identification.

11. A method for the delivery of voice messages to a voice service subscriber using voice commands, comprising:

(a) sensing a voice input command from the subscriber; and

(b) selecting at least one of a plurality of voice messages to deliver according to the voice input command sensed in step (a).

12. The method of claim 11, wherein the step (a) of sensing comprises a step of (c) converting the voice input command to digital voice data in an analog to digital converter.

13. The method of claim 12, further comprising a step (d) of storing the digital voice data.

14. The method of claim 13, further comprising a step of (e) discriminating at least one of a plurality of predetermined commands according to the digital voice data.

15. The method of claim 14, further comprising a step of (f) presenting the subscriber with voice message content according to the digital voice data.

16. The method of claim 15, further comprising a step of (g) presenting the subscriber with at least one voice command prompt to query voice input from the subscriber.

17. The method of claim 16, wherein the voice command prompt comprises a sequence of voice command prompts.

18. The method of claim 17, wherein the sequence of voice command prompts comprises a set of voice command prompts adaptively selected according to the digital voice data.

19. The method of claim 11, further comprising a step of (h) authenticating the subscriber for receipt of the voice messages.

20. The method of claim 19, wherein the step (h) of authenticating comprises at least one of prompting for PIN validation and voice identification.

## Abstract Of The Disclosure

The delivery of voice serve messages communicating financial, personal or other news telecasts may be customized according to the identity of the recipient and controlled by voice commands. A voice service bureau may generate voice messages for individual subscribers according to their preselected interest and needs. For instance, a financial bulletin may be abbreviated from "Your portfolio has declined by 5% this morning, do you want to sell" The subscriber may be identified using voice-activated PIN or other information, or be authenticated using voice recognition.

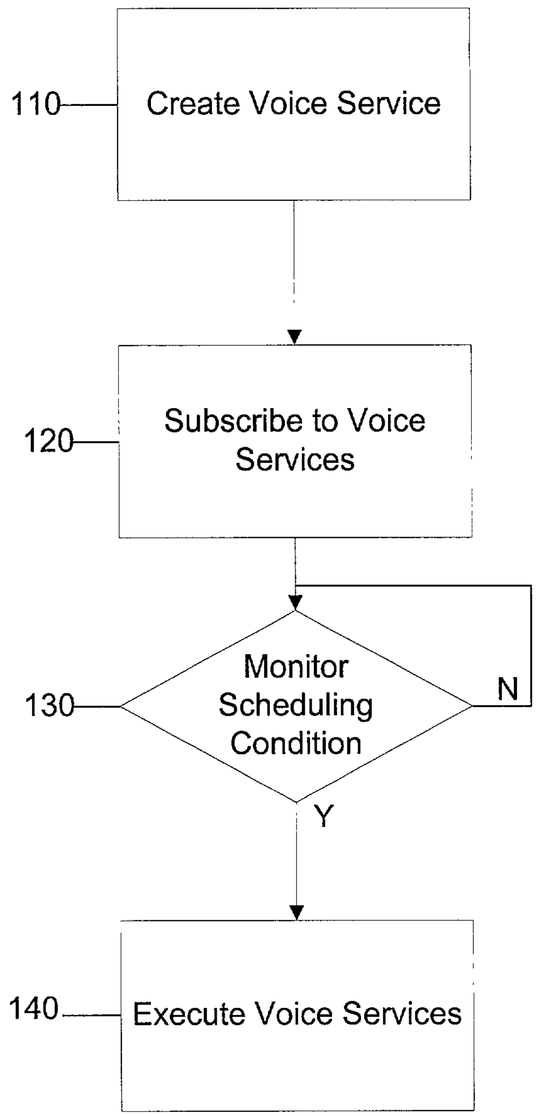
110
Create Voice Service
120
Subscribe to Voice Services
130
Monitor Scheduling Condition
N
Y
140
Execute Voice Services

**Figure 1**

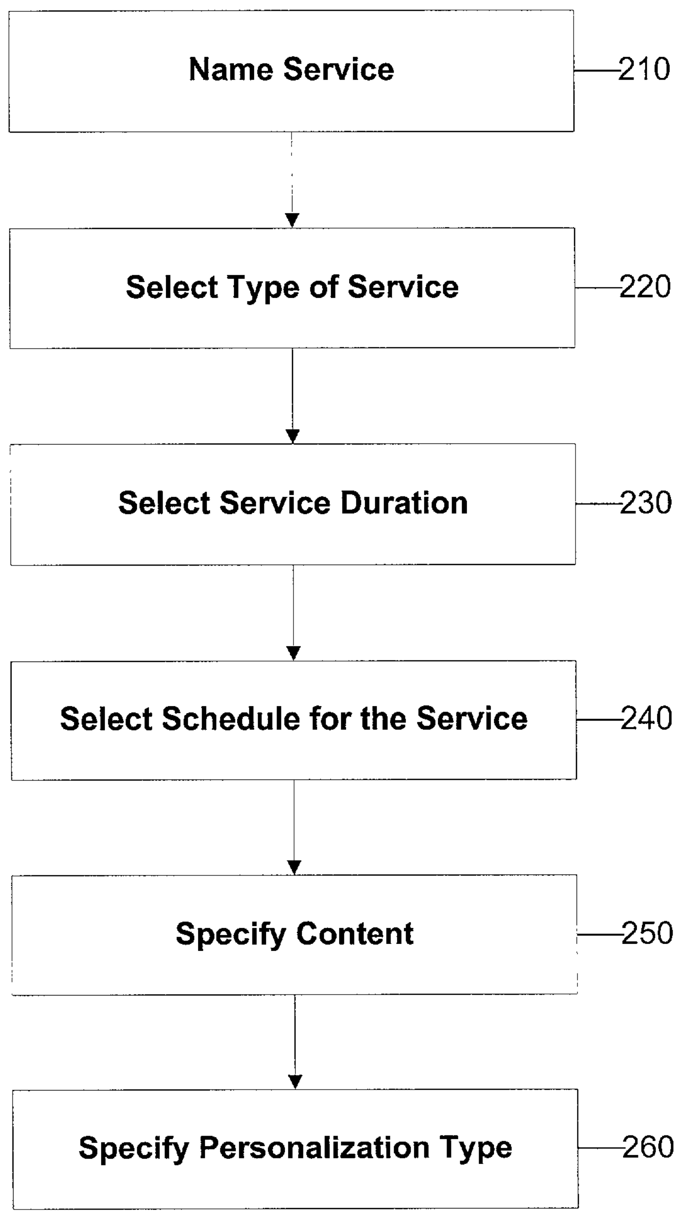
Name Service
210
Select Type of Service
220
Select Service Duration
230
Select Schedule for the Service
240
Specify Content
250
Specify Personalization Type
260

Figure 2

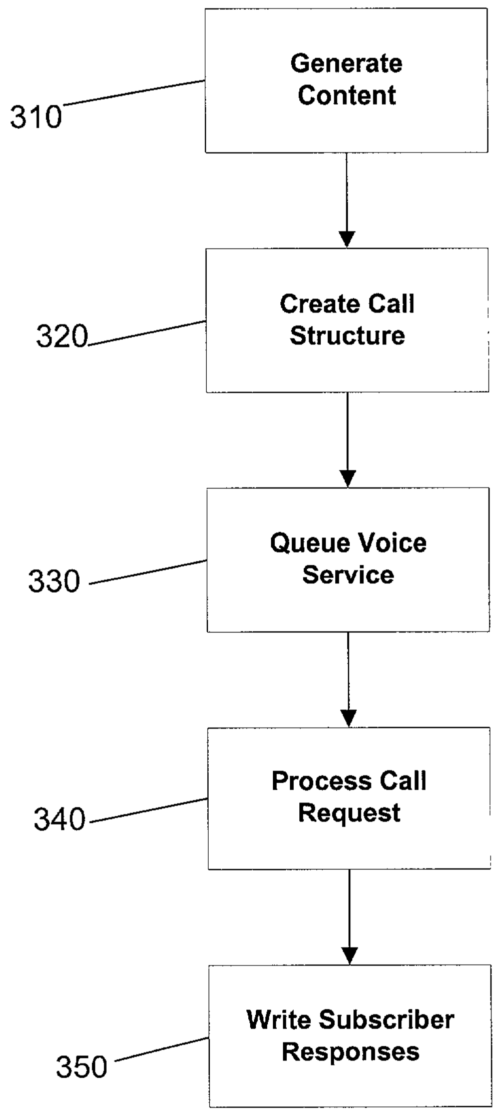
Generate Content
310
Create Call Structure
320
Queue Voice Service
330
Process Call Request
340
Write Subscriber Responses
350

**FIGURE 3**

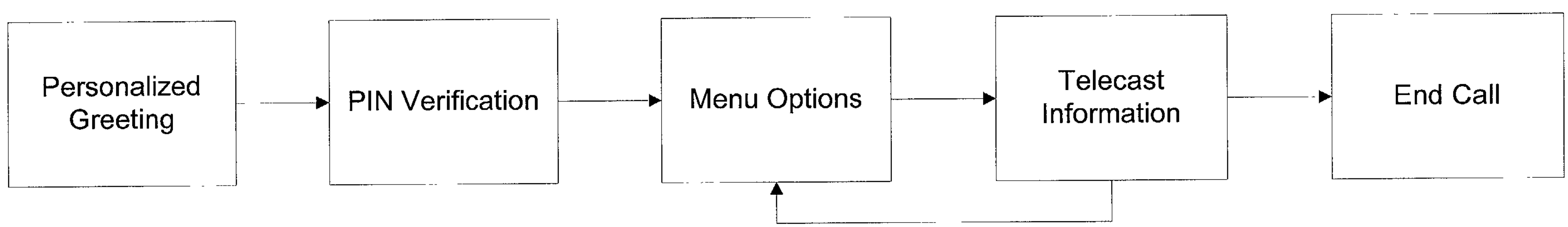
Personalized Greeting
PIN Verification
Menu Options
Telecast Information
End Call

FIGURE 3A

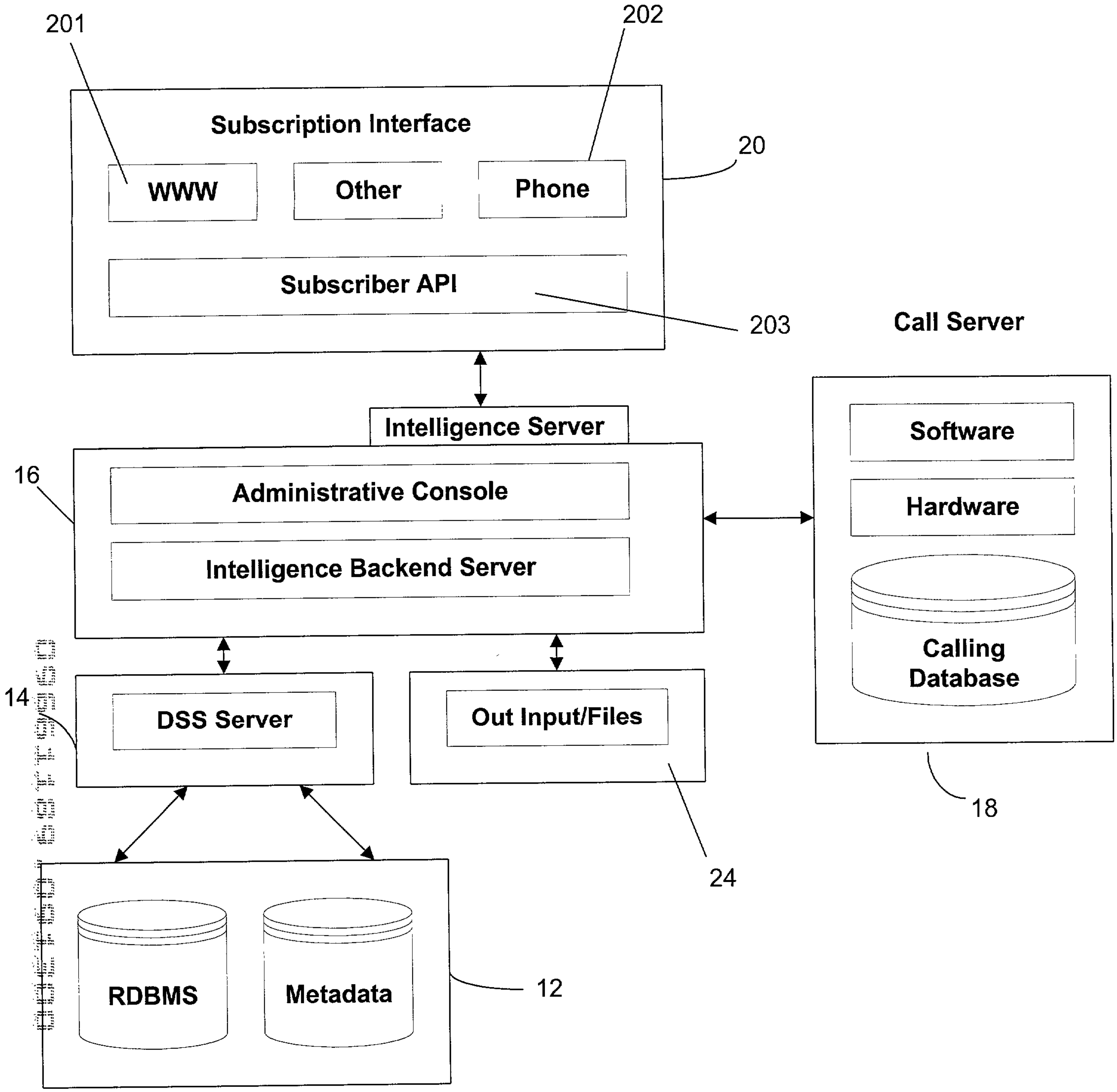
201
202
Subscription Interface
WWW
Other
Phone
20
Subscriber API
203
Call Server
Intelligence Server
Software
16
Administrative Console
Hardware
Intelligence Backend Server
Calling
Database
14
DSS Server
Out Input/Files
18
24
RDBMS
Metadata
12

FIGURE 4

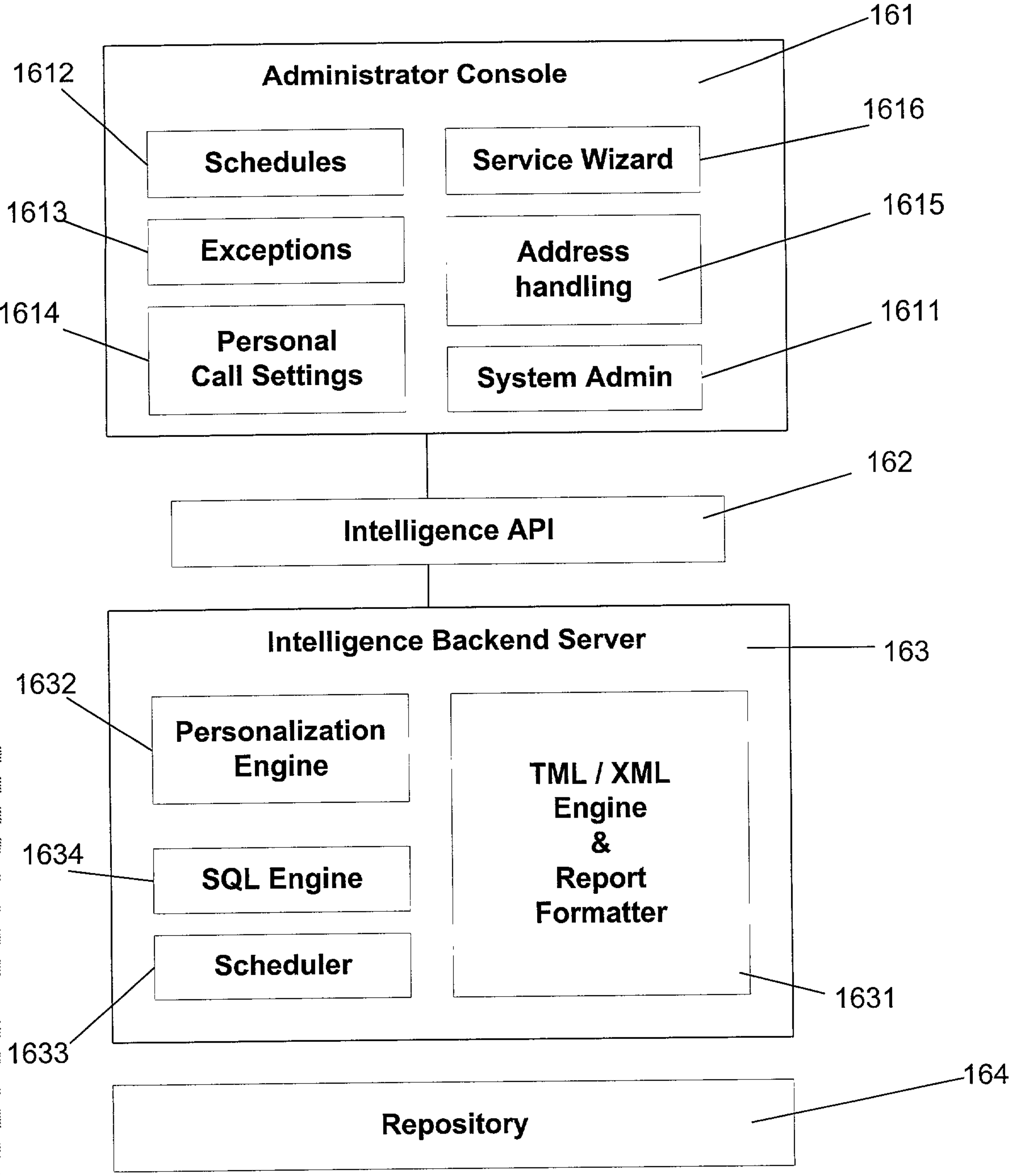
161
Administrator Console
1612
Schedules
Service Wizard
1616
1613
Exceptions
Address handling
1615
1614
Personal Call Settings
System Admin
1611
162
Intelligence API
Intelligence Backend Server
163
1632
Personalization Engine
TML / XML Engine & Report Formatter
1634
SQL Engine
Scheduler
1631
1633
164
Repository

FIGURE 5

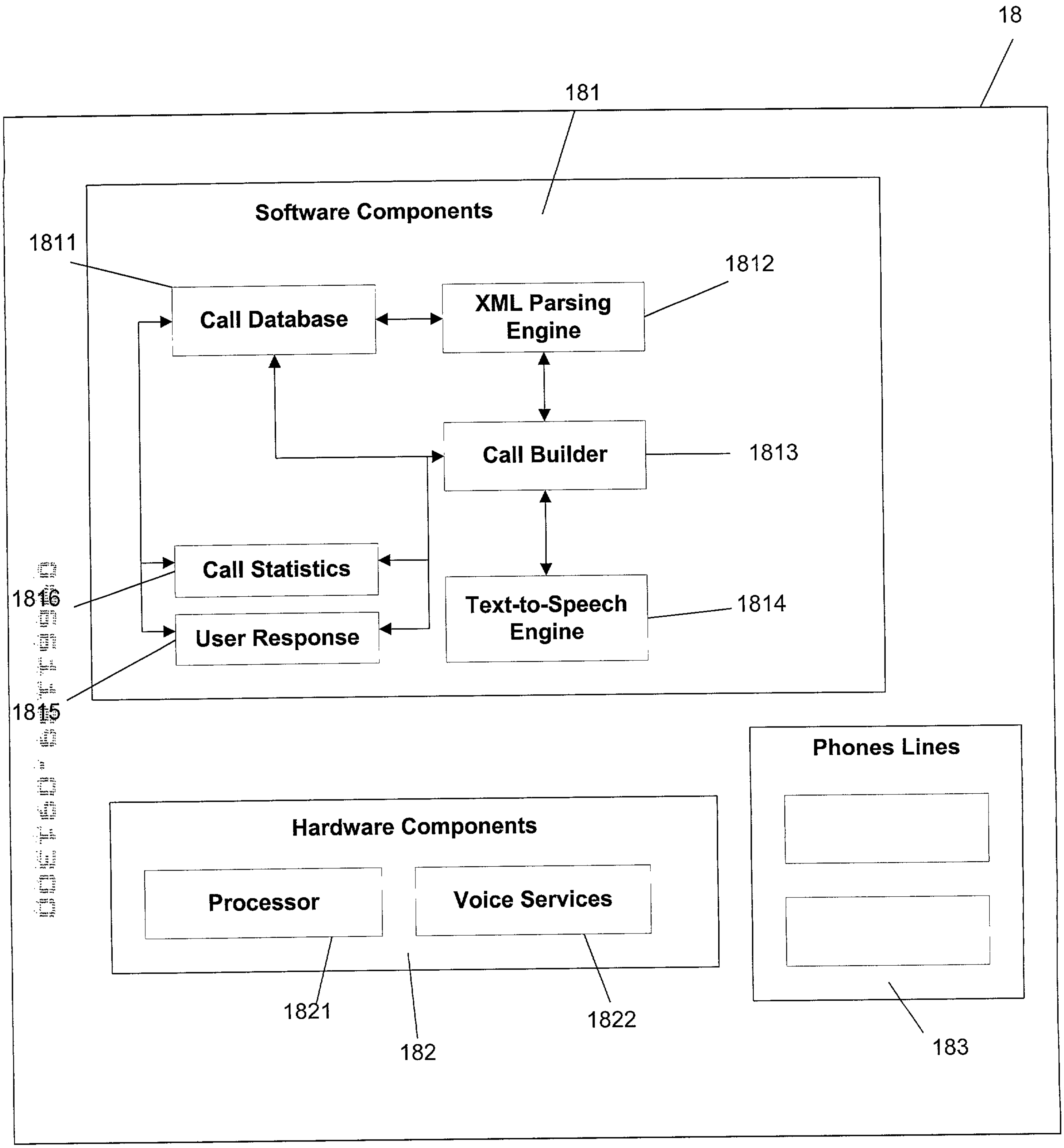
18
181
Software Components
1811
Call Database
XML Parsing Engine
1812
Call Builder
1813
Call Statistics
1816
User Response
1815
Text-to-Speech Engine
1814
Phones Lines
Hardware Components
Processor
Voice Services
1821
182
1822
183

**FIGURE 6**

Client-side Installation

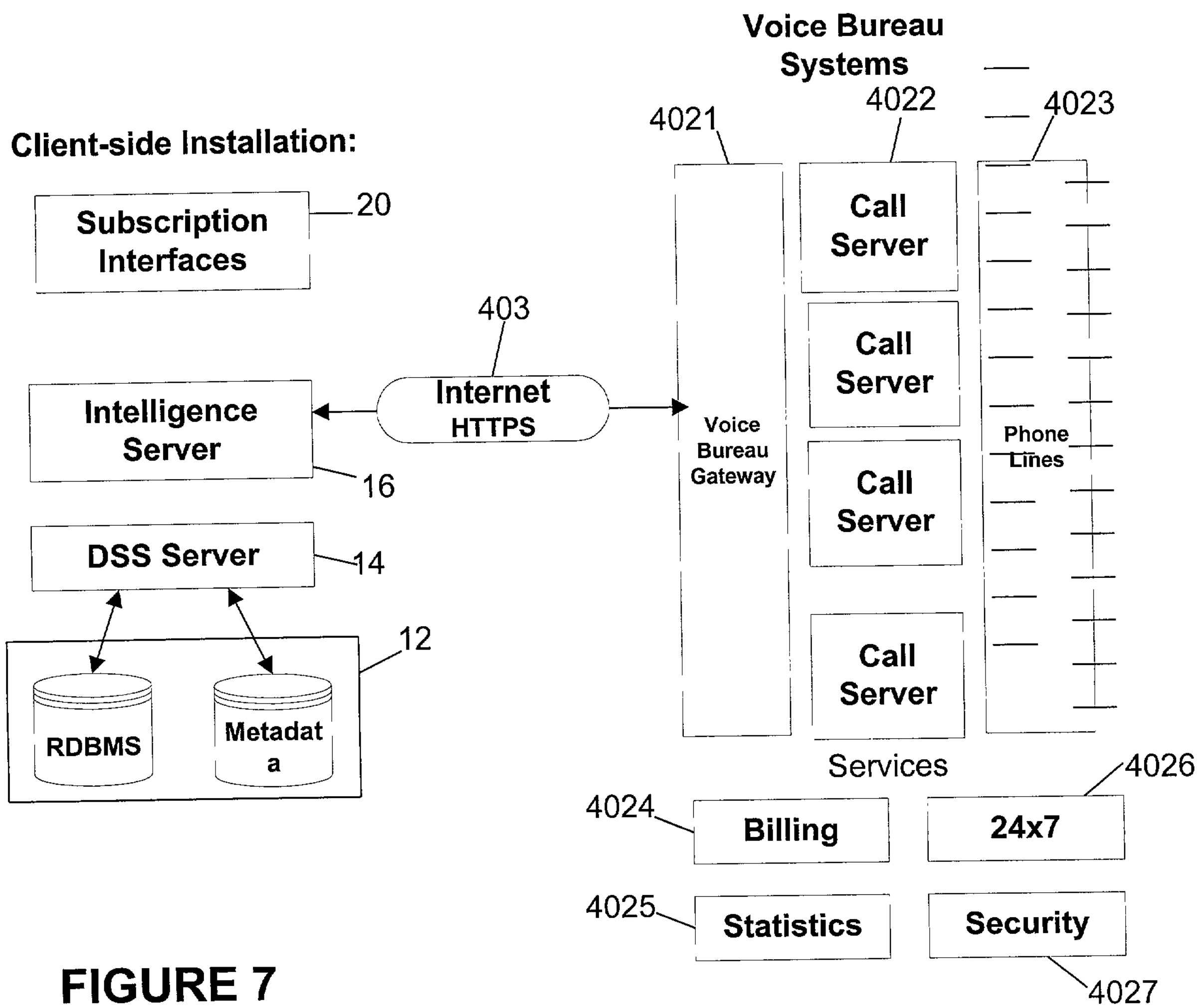


FIGURE 7

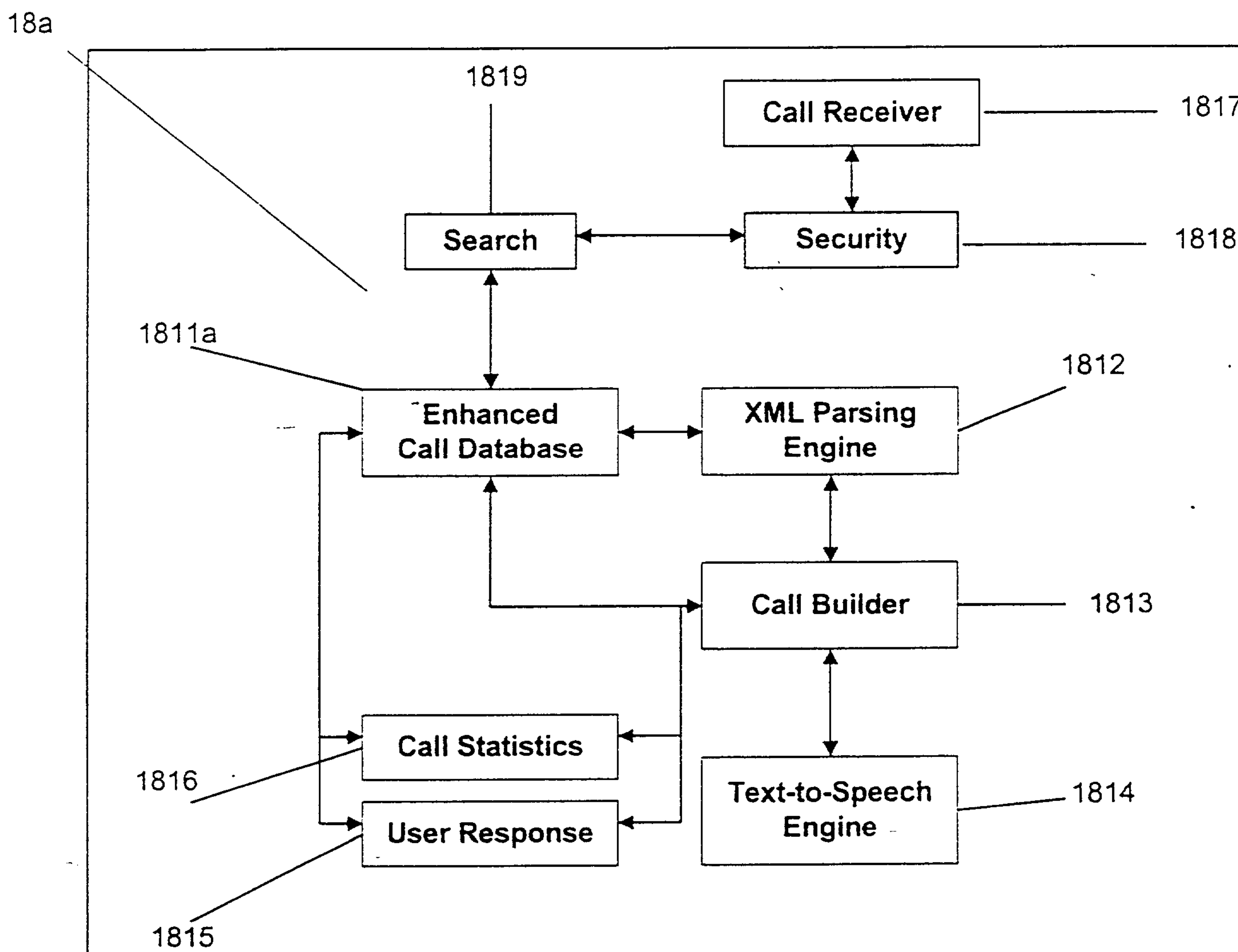
18a
1819
Call Receiver
1817
Search
Security
1818
1811a
1812
Enhanced Call Database
XML Parsing Engine
Call Builder
1813
Call Statistics
1816
User Response
Text-to-Speech Engine
1814
1815

FIGURE 8

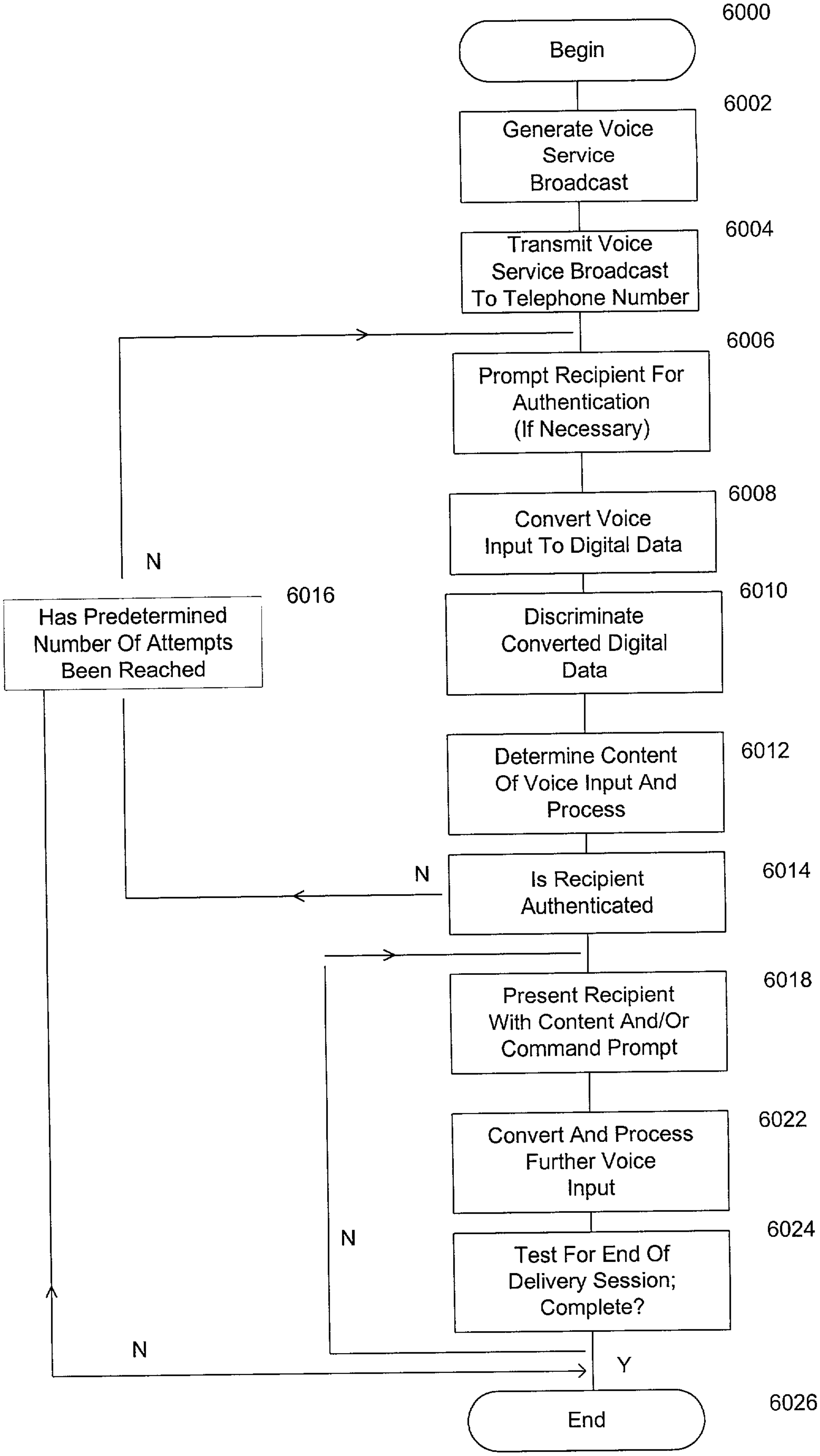
6000
Begin
6002
Generate Voice Service Broadcast
6004
Transmit Voice Service Broadcast To Telephone Number
6006
Prompt Recipient For Authentication (If Necessary)
6008
Convert Voice Input To Digital Data
6010
Discriminate Converted Digital Data
6012
Determine Content Of Voice Input And Process
6014
Is Recipient Authenticated
N
6016
Has Predetermined Number Of Attempts Been Reached
N
6018
Present Recipient With Content And/Or Command Prompt
6022
Convert And Process Further Voice Input
6024
Test For End Of Delivery Session; Complete?
N
Y
N
6026
End

FIGURE 9

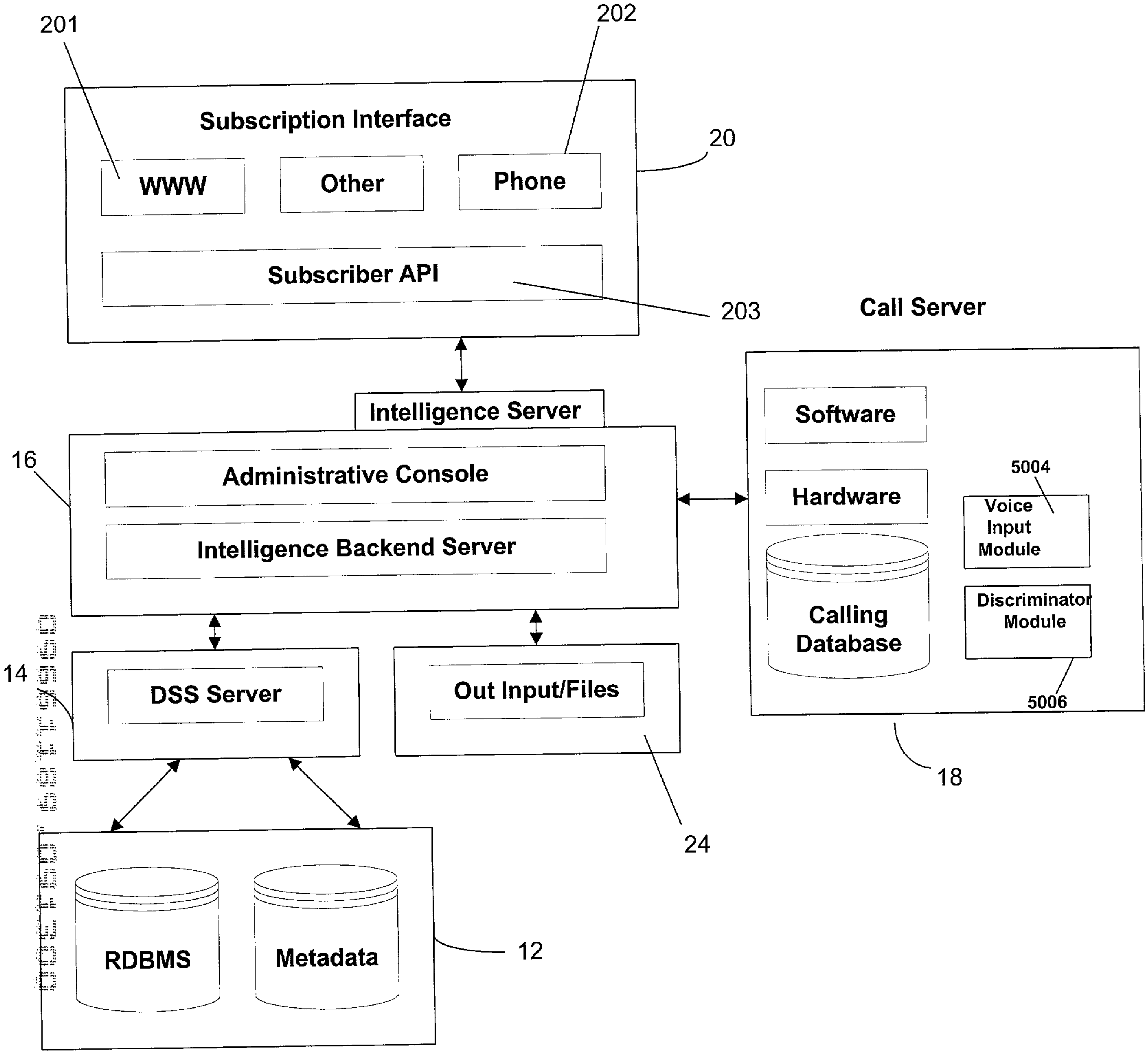
201
202
Subscription Interface
WWW
Other
Phone
20
Subscriber API
203
Call Server
Intelligence Server
Software
16
Administrative Console
Hardware
5004
Voice Input Module
Intelligence Backend Server
Calling Database
Discriminator Module
14
DSS Server
Out Input/Files
5006
18
24
RDBMS
Metadata
12

FIGURE 10

# JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenship are as stated below next to our names;

We believe that we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled **SYSTEM AND METHOD FOR VOICE-ENABLED INPUT FOR USE IN THE CREATION AND AUTOMATIC DEPLOYMENT OF PERSONALIZED, DYNAMIC AND INTERACTIVE VOICE SERVICES**, the specification of which

[X] is attached hereto.
[ ] was filed on ____________ as Application Serial Number ________________ and was amended on ______________________________
(if applicable)

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to in this declaration

We acknowledge the duty to disclose all information known to us to be material to the patentability of this application, as defined in 37 C.F.R. § 1.56.

We acknowledge the duty to disclose to the Office all information known to us to be material to patentability as defined in § 1.56, which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

## Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

| Country | Application Number | Date of Filing (day, month, year) | Date of Issue (day, month, year) | Priority Claimed Under 35 U.S.C. 119 |
|---|---|---|---|---|
| | | | | Yes [ ] No [ ] |
| | | | | Yes [ ] No [ ] |
| | | | | Yes [ ] No [ ] |
| | | | | Yes [ ] No [ ] |

## Prior United States Provisional Application(s)

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below

| Application Serial Number | Date of Filing (day, month, year) |
|---|---|
| 60/153,222 | 13 September 1999 |

## Prior United States Application(s)

We hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, we acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

| Application Serial Number | Date of Filing (day, month, year) | Status - Patented, Pending, Abandoned |
|---|---|---|
| | | |
| | | |

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith the following attorneys, their registration numbers being listed after their names:

**Thomas J. Scott, Jr., Registration No. 27,836; Stanislaus Aksman, Registration No. 28,562; James G. Gatto, Registration No. 32,694; Christopher C. Campbell, Registration No. 37,291; Henry C. Su, Registration No. 37,738; Brian M. Buroker, Registration No. 39,125; Charles F. Hollis, Registration No. 40,650; Jonathan D. Link, Registration No. 41,548; Kevin T. Duncan, Registration No, 41,495; George Georgellis, Registration No. 43,632; Stephen T. Schreiner, Registration No. 43,097; Christopher J. Cuneo, Registration No. 42,450; Raphael A. Valencia, Registration No. 43,216; Scott D. Balderston, Registration No. 35,436; Steven P. Klocinski, Registration No. 39,251; Yisun Song, Registration No. 44,487; Jennifer A. Albert, Registration No. 32,012; Kerry Owens, Registration No. 37,412; Milan M. Vinnola, Registration Number 45,979; Devin S. Morgan, Registration No. 45,562; Andrew J. Ririe, Registration No. 45,597; Carl Benson, Registration No. 38,378; Thomas E. Anderson, Registration No. 37,063; Thomas Blasey, Registration No. 33,475; Robin Clark, Registration No. 40,956; René Vazquez, Registration No. 38,647 and David M. Huntley, Registration No. 40,309.**

All correspondence and telephone communications should be addressed to Hunton & Williams, 1900 K Street, N.W., Suite 1200, Washington, D.C. 20006-1109, telephone number (202) 955-1500, which is also the address and telephone number of each of the above listed attorneys.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon

| | | | |
|---|---|---|---|
| Full Name of First Inventor | **EBERLE**<br>Family Name | **HANNES**<br>First Given Name | Second Given Name |
| Residence | **5016 N. 6th Street, Arlington, VA 22203** | | |
| Citizenship | **Germany** | | |
| Post Office Address | **5016 N. 6th Street, Arlington, VA 22203** | | |
| Signature ________________ | | Date ________________ | |

| | | | |
|---|---|---|---|
| Full Name of Second Inventor | **LEON**<br>Family Name | **CHRISTOPHER**<br>First Given Name | **S.**<br>Second Given Name |
| Residence | **2443 Ontario Rd. N.W., Washington, DC 20009** | | |
| Citizenship | **United States of America** | | |
| Post Office Address | **2443 Ontario Rd. N.W., Washington, DC 20009** | | |
| Signature ________________ | | Date ________________ | |

| | | | |
|---|---|---|---|
| Full Name of Third Inventor | **MAASS**<br>Family Name | **BODO**<br>First Given Name | Second Given Name |
| Residence | **5016 N. 6th Street, Arlington, VA 22203** | | |
| Citizenship | **Germany** | | |
| Post Office Address | **5016 N. 6th Street, Arlington, VA 22203** | | |
| Signature ________________ | | Date ________________ | |

| | | | |
|---|---|---|---|
| Full Name of Fourth Inventor | **PATNAIK** Family Name | **ANURAG** First Given Name | Second Given Name |
| Residence | **850 N. Randolph Street, Apt. 2120, Arlington, Virginia 22201** | | |
| Citizenship | **India** | | |
| Post Office Address | **850 N. Randolph Street, Apt. 2120, Arlington, Virginia 22201** | | |

Signature ______________________________ Date ____________________

| | | | |
|---|---|---|---|
| Full Name of Fifth Inventor | **SANTA ANA** Family Name | **ALBERTO** First Given Name | Second Given Name |
| Residence | **2903 Wickersham Way, Apt. #T2, Falls Church, VA 22042** | | |
| Citizenship | **Mexico** | | |
| Post Office Address | **2903 Wickersham Way, Apt. #T2, Falls Church, VA 22042** | | |

Signature ______________________________ Date ____________________

| | | | |
|---|---|---|---|
| Full Name of Sixth Inventor | **ZIRNGIBL** Family Name | **MICHAEL** First Given Name | Second Given Name |
| Residence | **1507 30th Street N.W., Apt. 3, Washington, DC 20007** | | |
| Citizenship | **Germany** | | |
| Post Office Address | **1507 30th Street N.W., Apt. 3, Washington, DC 20007** | | |

Signature ______________________________ Date ____________________

**HUNTON & WILLIAMS**
1900 K Street, N.W., Suite 1200
Washington, D.C. 20006-1109
Tel: (202) 955-1500